

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 3906T001 – Mechatronika

**Tvorba algoritmů pro řídicí jednotku
průmyslového robota**

**Development and implementation of
algorithms for industrial robot control unit**

Diplomová práce

Autor: **Bc. Michal Novotný**
Vedoucí práce: Ing. David Lindr
Konzultant: Ing. Martin Diblík, Ph.D.

Rozsah práce: 55 stran včetně příloh
21 obrázků
4 tabulky
1 vázaná příloha
1 volná příloha CD

V Liberci 29. 5. 2009

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum: 29.5.2009

Podpis:

Poděkování

Touto psanou formou bych rád poděkoval svému konzultantovi Ing. Davidu Lindrovi za cenné rady, podnětné připomínky a poskytnuté konzultace z oblasti programování řídicích jednotek průmyslových robotů a teorie řízení robotů.

Rovněž bych chtěl poděkovat i ostatním členům z Ústavu mechatroniky a technické informatiky za informace ohledně metodiky zpracování diplomové práce.

Liberec, květen 2009

Michal Novotný

Anotace

Diplomová práce se zabývá spoluprací na vývoji řídicí jednotky pro šestiosého robota KUKA VK10/15. Značná část práce je věnována tvorbou, implementací a testováním algoritmu přímé a inverzní úlohy pro transformaci souřadnic šestiosého angulárního mechanismu do kartézských souřadnic v prostředí MATLAB. Následně se zabývá jeho konverzí a implementací do software Simotion Scout určeného pro zvolenou řídicí jednotku SIMOTION D435. Dále navazuje testováním a implementací algoritmů pro lineární interpolaci a pozdější úpravou algoritmu zajišťující ztotožnění výsledných řešení mezi oběma vývojovými nástroji. Pro co nejrychlejší a nejpresnější vykonávání pohybů robota se práce zabývá i nastavením regulačních smyček pro jednotlivé rotační osy této angulární struktury.

Klíčová slova: pohon, přímá a inverzní úloha, robot, regulátor, řízení po přímce

Annotation

Diploma thesis deals with cooperation in creating of control unit for six-angle robot KUKA VK10/15. Great deal of work is dedicated in production, implementation and testing of algorithm forward and inverse kinematic for transformation of six-angle coordinated angular mechanism into cartesian coordinates at MATLAB. After deals with conversion into Simotion Scout designed for chosen control unit SIMOTION D435. Then deals with testing and implementation of algorithm for linear interpolation and later modification handling to identify of solutions between development tools. For fastest and precise movement deals with parametrization of regulation loop for all motors.

Keywords: drive, forward and inverse kinematic, robot, regulator, linear interpolation control

Obsah

Seznam obrázků:	9
Seznam tabulek:	9
Seznam zkratek a pojmů:	10
1. Úvod	11
2. Matematický aparát pro řízení robotů	13
2.1 Homogenní souřadnice	13
2.2 Transformace mezi souřadnými systémy	14
2.3 Přímá úloha	15
2.4 Inverzní úloha	15
3. Model řízení robotů	16
4. Současný stav robotiky na TUL	18
4.1 Popis robota	18
4.2 Řídící jednotka Simotion D435 a Sinamics S120	20
4.3 Ovládací panel	21
4.4 Vzájemná komunikace použitých HW zařízení	23
5. Realizace algoritmů	25
5.1 Algoritmus přímé úlohy	26
5.1.1 Realizace v MATLABU	26
5.2 Algoritmus inverzní úlohy	29
5.2.1 Realizace v MATLABU	29
5.3 Řízení po přímce	32
5.4 Testování algoritmů	33
5.5 Konverze algoritmu do prostředí Simotion Scout	33
5.6 Chyby a nepřesnosti algoritmu	35
5.7 Měření hodnoty	37
6. Nastavení regulátorů	39
6.1 Teorie regulace	39
6.1.1 Typy regulátorů	39
6.1.2 Zásady regulace	40
6.1.3 Kaskádní regulace	41
6.2 Nastavení parametrů regulátorů	42
7. Závěr	51
Literatura	53

Seznam obrázků:

Obr. 2.1 Popis vektorů matice pomocí souřadných systémů	14
Obr. 3.1 Popis rotačních os robota.....	16
Obr. 4.1 Popis rotačních os robota.....	19
Obr. 4.2 Konstrukční schéma průmyslového robota z bokorysu	19
Obr. 4.3 Konstrukční schéma průmyslového robota z půdorysu	20
Obr. 4.4 Řídící jednotka Simotion D435 a frekvenční měnič Sinamics S120.....	21
Obr. 4.5 Ovládací panel	22
Obr. 4.6 Prostředí <i>NetPro</i> pro definice komunikace mezi hardwarovými moduly	24
Obr. 5.1 Grafické znázornění natočení mechanismů robota v prostoru	28
Obr. 5.2 Grafické znázornění polohy mechanismů základní polohy v záporném smyslu orientace čelistí chapadla ve 2D	29
Obr. 5.3 Grafické znázornění polohy mechanismů základní polohy v kladném smyslu čelistí chapadla ve 2D.....	32
Obr. 6.1 Přejížděvací charakteristika dvou dynamických systémů	40
Obr. 6.2 Blokové schéma kaskádního regulačního obvodu.....	41
Obr. 6.3 Přejížděvací charakteristika proudové smyčky 4 osy před optimalizací	44
Obr. 6.4 Přejížděvací charakteristika proudové smyčky 4 osy po optimalizaci	45
Obr. 6.5 Frekvenční charakteristika proudové smyčky 4 osy před ladící procedurou.....	46
Obr. 6.6 Frekvenční charakteristika proudové smyčky 4 osy po ladící proceduře	46
Obr. 6.7 Přejížděvací charakteristika rychlostní smyčky 4 osy před optimalizací	47
Obr. 6.8 Přejížděvací charakteristika rychlostní smyčky 4 osy po optimalizaci	48
Obr. 6.9 Frekvenční charakteristika rychlostní smyčky 4 osy před ladící procedurou.....	48
Obr. 6.10 Frekvenční charakteristika rychlostní smyčky 4 osy po ladící proceduře	49

Seznam tabulek:

Tab. 1 Parametry regulátorů před optimalizací.....	44
Tab. 2 Parametry regulátorů po optimalizaci.....	44
Tab. 3 Parametry synchronního motoru 4 osy	50
Tab. 4 Porovnání regulačních vlastností před a po ladící proceduře.....	50

Seznam zkratk a pojmů:

ALM	Active line module
D435	řídící jednotka Simotion D435
SINAMICS S120	frekvenční měnič
homing	výchozí pozice osy
PROFIBUS	průmyslová komunikační sběrnice
PC	počítač (Personal computer)
Ethernet	lokální komunikační síť
ROM	paměť pouze pro čtení (read only memory)
CU320	řídící jednotka SINAMICS
PROFIBUS	průmyslová komunikační sběrnice
MPI	rozhraní pro komunikaci s D435
SM	Sensor modul
MM	Motor modul

1. Úvod

Laboratoř Ústavu mechatroniky a technické informatiky je vybavena průmyslovým robotem KUKA VK10/15, který slouží pro účely vývoje řídicích algoritmů v programovacím prostředí Simotion Scout. Robot byl používán firmou Škoda auto a.s., kde byl vyřazen z provozu díky jeho vadné řídicí jednotce. Fakulta se rozhodla robota odkoupit a jeho originální řídicí systém nahradit novým. Byl proto vybrán kompatibilní modulární celek od firmy Siemens, který byl pro tuto aplikaci nejvhodnější. Je složen z jednotky Simotion D435 (dále jen D435) a frekvenčního měniče SINAMICS S120 (dále jen S120). D435 obsahuje interpolační nástroje pro maximálně čtyř-osé mechanismy. Proto bylo nutné prostředky pro řízení šestiosého mechanismu vyvinout, zalgoritmizovat a do řídicího systému implementovat. Rovněž bylo potřeba systém vybavit interpolačním nástrojem umožňující vést koncový člen manipulátoru po základních i obecných křivkách.

Přímá úloha byla vypočítána pomocí Denavit-Hartenbergovy maticové metody spočívající ve vytvoření transformační matice mezi dvěma sousedními souřadnými systémy v univerzálním tvaru, tak aby bylo možno provést snadnou algoritmizaci a naprogramování výpočtu této úlohy. Inverzní úlohu se podařilo vyjádřit analyticky a nemusí být tedy počítána numericky, což je výhodou, především z hlediska výpočetního výkonu. Z celkového matematického odvození obou úloh jsou v algoritmu použity pouze koncové vztahy, ze kterých se po dosazení všech výrazů získá konkrétní výsledek (žádaný úhel, složka polohového vektoru či matice orientace). Algoritmizace matematických vztahů byla nejdříve provedena v prostředí MATLAB, ve kterém je možné algoritmy ověřovat. Po ověření byl vytvořen algoritmus ve vývojovém prostředí Simotion Scout, který je určen přímo pro programování a konfigurování výše uvedených jednotek, tj. D435 a S120. Obě jednotky tvoří modulární systém, který jednak zajišťuje patřičné vykonání definovaného algoritmu, ale také dovoluje řízení na výkonové úrovni, tedy na úrovni pohonů. Jeho výhodou je také řízení pohybů servopohonů podle daných křivek a víceosé řízení polohy.

Diplomová práce částečně navazuje na projekt [9], který se zabýval kalibrací odměřovacího systému průmyslového robota. I v tomto případě je nutné před prvním uvedením do provozu zajistit výchozí polohu os robota.

V algoritmu je rovněž implementována funkce umožňující pohyb koncového

bodů robota po přímce. Také tato funkce byla nejprve vyvinuta a ověřena v MATLABu a později byla přidána do Simotion Scout. Blíže je popsána v kapitole 5.3. Aplikovat ji lze v průmyslové automatizaci, kdy je potřeba dodržet určitý technologický postup, například při svařování.

U všech rotačních mechanismů je nutné zajistit vykonávání pohybů co nejpřesněji a nejrychleji. Právě díky Simotion Scout je také možné definovat parametry jednotlivých regulačních struktur, proto se činnost soustředila i do problematiky o návrhu parametrů regulátorů proudové, rychlostní a polohové smyčky.

K pohodlnému a přehlednému ovládání robota byla vyvinuta aplikace pro mobilní panel, který dovoluje zadávat příkazy o poloze v kartézském souřadném systému tak i definovat jednotlivé úhlové natočení rotačních os robota.

2. Matematický aparát pro řízení robotů

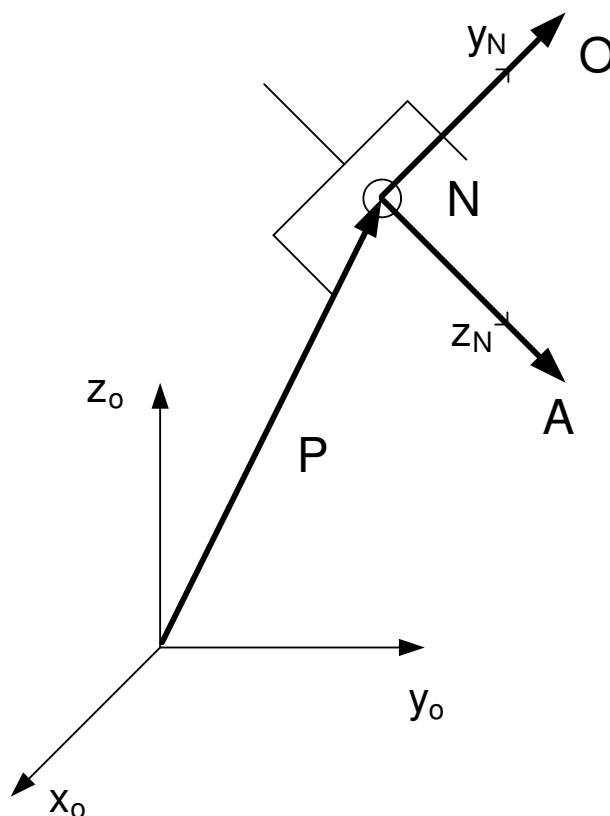
Důležitou částí analýzy průmyslových robotů je získání matematického modelu, který poskytuje potřebné matematické vztahy. Při řízení pohybů robota je možné celou kinematickou strukturu rozdělit na jednotlivé části a ty řešit samostatně. U nich je potom potřeba zjistit polohu a orientaci v prostoru nebo určit konkrétní pozice v závislosti na vstupních proměnných. Na základě tohoto požadavku je možné provést analýzu, kterou lze rozdělit na přímou a inverzní úlohu, jejíž popis bude uveden dále. Před tím však zavedme některé základní pojmy.

2.1 Homogenní souřadnice

Kinematiku mechanismů lze v robotice popsat pomocí matic, které se skládají ze sloupcových vektorů. Tyto vektory mají místo třírozměrných souřadnic zavedeny souřadnice čtyři, kde čtvrtá souřadnice bývá zavedena z formálních důvodů. Pouze zjednodušuje implementaci algoritmu pokud se provádí několik transformací v souřadných systémech a určuje zda se jedná o souřadnice vektoru (její hodnota je 0) nebo bodu (její hodnota je 1). Základní matici, která kromě polohy bodu určuje i orientaci je popsána vztahem (1). Je složena ze zmíněných čtyřrozměrných vektorů, z nichž první tři tvoří souřadný systém a čtvrtý sloupec určuje polohu od vztažného souřadného systému.

$$A = \begin{bmatrix} N_x & O_x & A_x & P_x \\ N_y & O_y & A_y & P_y \\ N_z & O_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Třetí sloupec matice představuje směr zaměření koncového bodu (chapadla), je to tzv. Acces vector, druhý sloupec se nazývá Orientation vector, jehož hodnoty vyjadřují smysl směru čelistí chapadla. První sloupec nemá svůj zvláštní název a pouze vhodně doplňuje matici. Vznikne kartézským součinem druhého a třetího sloupce matice, tak aby výsledná soustava byla pravotočivá a zároveň ortonormální ($N=O \times A$). Pro snazší představu je na obr. 2.1 uveden koncový efektor s označením významu jednotlivých vektorů pomocí souřadných systémů.



Obr. 2.1 Popis vektorů matice pomocí souřadných systémů

2.2 Transformace mezi souřadnými systémy

Do jednotlivých článků kinematické struktury se zavedou lokální souřadné systémy. Mezi globálním, který bývá zvolen v pracovním prostoru robota a koncovým, tj. tam kde se nachází koncový bod se pomocí maticového počtu provádí transformace. Výsledná transformační matice je potom složena z násobení těchto lokálních matic v pořadí, které odpovídá pořadí souřadných systémů od globálního až po koncový souřadný systém. Tyto matice jsou buď translační nebo rotační.

Transformační matice lze u složitějších kinematických struktur získat některými metodami, z nichž neznámější je asi podle Denavit-Hartenbergovy konvence. Tato maticová metoda vychází z přiřazování souřadnicových systémů každému z článků. Transformační vztahy jsou potom založeny na fiktivních pohybech souřadnicových systémů tak, aby došlo k jejich sjednocení. Více je tato metoda popsána např. v [13].

2.3 Přímá úloha

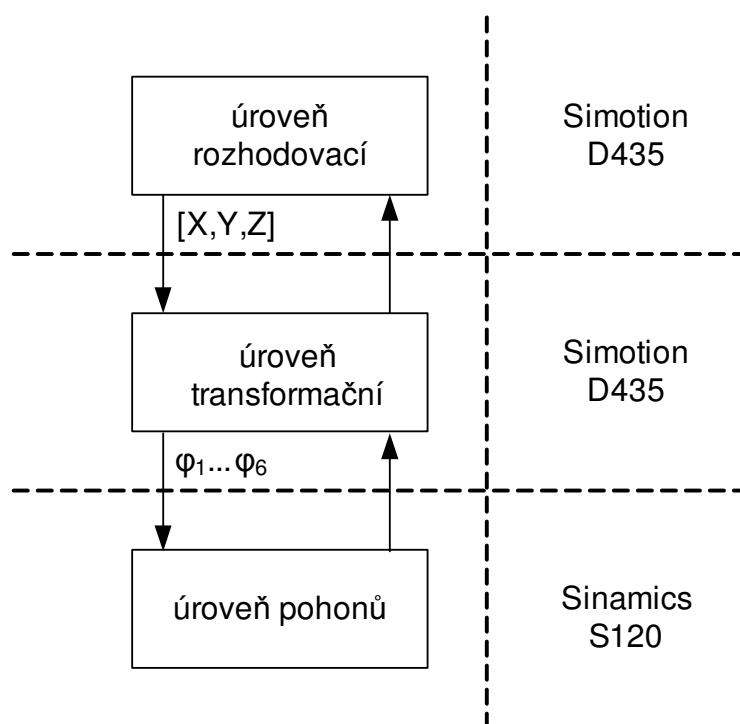
K popisu kinematické struktury se používají dvě úlohy. Pokud jde o přímou úlohu, jsou známy jednotlivé souřadnice kinematických členů a hledáme orientaci a polohu koncového bodu. Tuto úlohu lze řešit pomocí goniometrických vztahů mezi jednotlivými články, nebo přes lokální souřadné systémy a transformační maticemi mezi nimi. Jak již bylo zmíněno, Denavit-Hartenbergovu metodu lze využít pro sestavení transformačních matic. Jejím účelem je vytvoření transformační matice mezi 2 sousedními lokálními souřadnými systémy v univerzálním tvaru, aby bylo možno provést snadnou algoritmizaci a naprogramování výpočtu přímé úlohy. Výstupem této úlohy je právě matice (1).

2.4 Inverzní úloha

Druhým typem popisu je inverzní úloha. Narozdíl od přímé úlohy je vstupním parametrem matice (1) a výsledkem jsou potom jednotlivé souřadnice kinematických členů. Řešení této úlohy je podstatně obtížnější a jsou založené buď na numerických nebo vektorových metodách, které jsou patřičně popsány v [13]. V našem případě se vztahy popisující inverzní úlohu podařilo získat analyticky, proto není nutné provádět výpočty numericky a tím je i ušetřeno výpočetního výkonu. Výsledná řešení této úlohy jsou nejednoznačná a jejich konkrétní počet může být variabilní.

3. Model řízení robotů

Při realizaci řídicího systému robota bylo požadavkem, aby splňoval víceúrovňový model řízení a bylo možné zasahovat do různých vrstev řízení. Tyto vrstvy lze popsat pomocí modelu, jehož jednotlivé vrstvy jsou hierarchicky řazeny podle funkcí. Sousední vrstvy si vzájemně vyměňují potřebné informace, což je předpokladem pro správnou funkci celého řídicího systému. Na obr. 3.1 je zaneseno blokové schéma řídicího systému členěného do příslušných vrstev.



Obr. 3.1 Popis rotačních os robota

- **úroveň pohonů**

Tato nejnižší úroveň zajišťuje provedení všech pohybů podle programu. Přímo řídí jednotlivé synchronní motory a využívá i zpětné informace o polohách.

- **úroveň transformační**

Nad nejnižší vrstvou se nachází transformační. Realizuje konverze kartézských souřadnic $[x,y,z]$, které jsou vztaženy vůči globálnímu souřadnému systému na požadované úhlové natočení jednotlivých kinematických článků (rotačních os). Vstupním parametrem je žádaná trajektorie pohybu koncového bodu. V této vrstvě se řeší jednak interpolace dané trajektorie, ale i zmíněné transformace. Výstupními

parametry jsou hodnoty úhlových natočení jednotlivých kinematických členů.

- **úroveň rozhodovací**

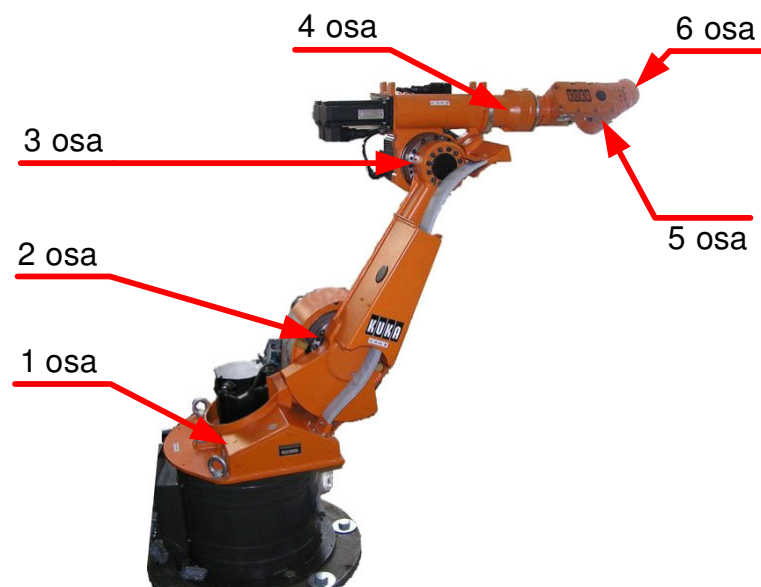
Nejvyšší vrstva vyhodnocuje požadavky od uživatele a provádí výpočty ramene robota po optimální dráze. Hodnotí se jednak podle míst, kterými nelze vést cílený pohyb a dle dalších kritérií např. z hlediska energetické náročnosti, apod. Do této vrstvy musí být také zavedeny signály ze snímačů, podle kterých je možné provádět patřičná rozhodnutí.

4. Současný stav robotiky na TUL

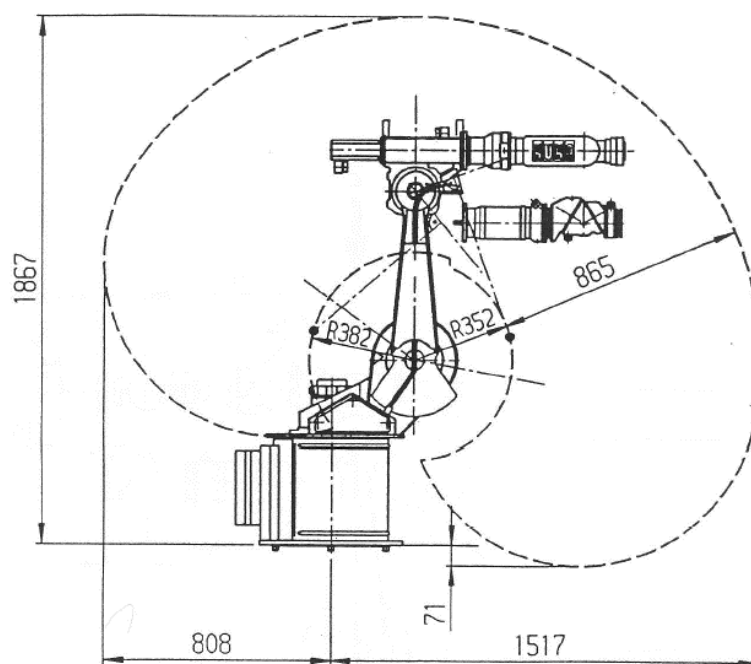
Fakulta mechatroniky, informatiky a mezioborových studií disponuje několika průmyslovými roboty od firmy ABB, které využívají ke svému řízení standardní řídicí systémy. Ty nedovolují aplikovat složitější algoritmy řízení, neumožňují přímý zásah do regulačních smyček a jejich řídicí algoritmus zůstává neveřejný. Z tohoto důvodu bylo potřeba vyvinout systém jehož struktura bude otevřená a bude podporovat širší interakci mezi jednotlivými funkčními bloky. Za tímto účelem byla laboratoř vybavena řídicím systémem od společnosti SIEMENS, který jednotlivé požadavky splňuje. Jeho konkrétní části spolu s průmyslovým robotem jsou detailně popsány na následujících řádkách.

4.1 Popis robota

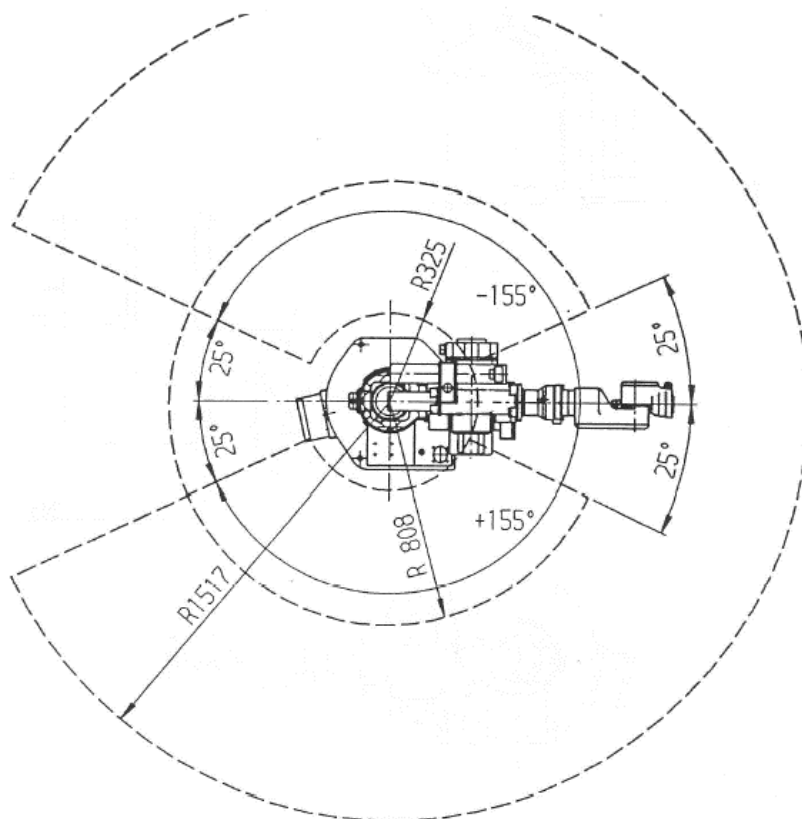
Průmyslový robot KUKA VK10/15 je osazen šesti synchronními motory. Každý motor je opatřen třípólovým resolverem, který slouží k odměřování polohy. Mimo jiné jsou motory vybaveny také bezpečnostními brzdami, která zabraňují samovolnému pootočení jednotlivých os. Celková struktura má všechny osy rotační a jedná se tedy o angulární koncepci. Na obr. 4.1 je uvedena fotografie robota s číselným popisem jednotlivých os. Počáteční souřadný systém se nachází v místě upevnění mezi mobilním rámem, který zajišťuje stabilitu celého mechanismu proti pádu a robotem. K němu je vztažena pozice ostatních souřadnic v prostoru. Koncový souřadný systém je tvořen šestou osou a na ní lze připevnit koncové chapadlo. Souřadnice tohoto koncového bodu jsou omezeny pracovním prostorem robota. Do tohoto omezení patří jednak rozměry manipulátoru ale i konstrukční minimální a maximální úhlové natočení konkrétní osy. Nelze tedy zaujmout libovolnou prostorovou pozici při dané orientaci. S těmito omezeními se také v algoritmu musí počítat. Na osách číslo 1 až 3 jsou za motorem převodovky jejichž převodový poměr je navržen do pomalu s poměrem 1:33. U zbývajících os nebyl převodový poměr předem znám, proto byl změřen experimentálně. Konstrukční schéma angulární struktury z bokorysu je zachyceno na obr. 4.2 a z půdorysu na obr. 4.3. Schémata znázorňují jednak konstrukční rozměry ale i úhlová omezení.



Obr. 4.1 Popis rotačních os robota



Obr. 4.2 Konstrukční schéma průmyslového robota z bokorysu

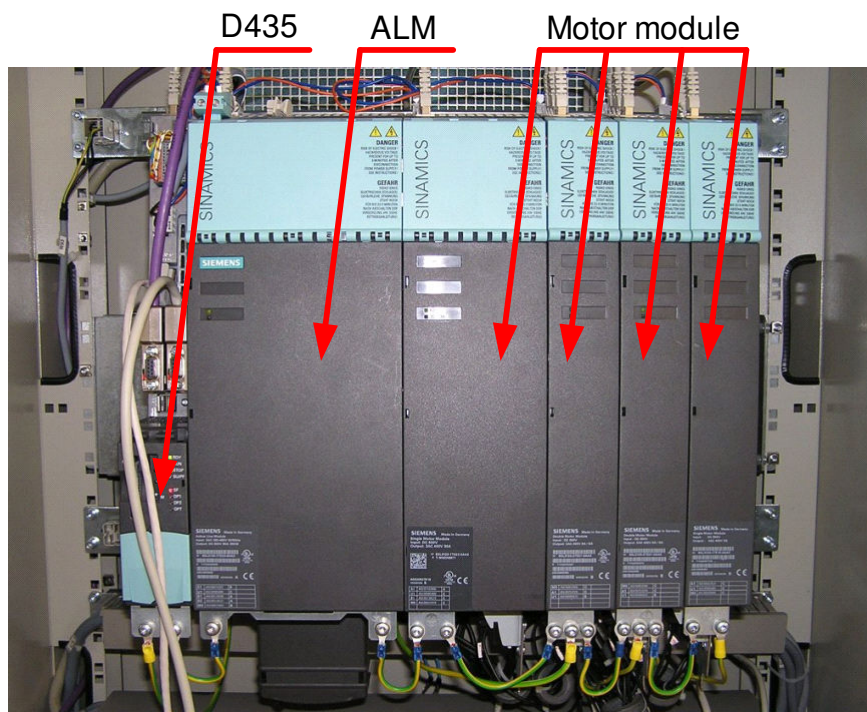


Obr. 4.3 Konstrukční schéma průmyslového robota z půdorysu

4.2 Řídící jednotka Simotion D435 a Sinamics S120

Jak již bylo výše zmíněno, řídicí systém nebylo kvůli závažné poruše možno použít a neumožňoval jako většina běžných systémů otevřenost. Z tohoto důvodu bylo přistoupeno k variantě náhrady, která nabízí otevřenost a umožňuje přístupy a zásahy do daných vrstev řízení, které byly popsány v kapitole 3. Pro tyto účely byl navržen modulární řídicí systém, jež se skládá z jednotlivých funkčních bloků. Ty jsou rozděleny podle víceúrovňového modelu řízení, který byl popsán v kapitole 3 a umožňují tedy jednak řízení pohybů robota, ale i realizaci vyšších algoritmů. Jako řídicí jednotka jednotlivých os robota byla zvolena S120. Ta je složena z řídicí jednotky pohonů CU320 (dále jen CU320), která řídí funkce usměrňovače a výkonových střídačů jednotlivých os. K nezbytné funkci celé S120 byla vybrána řídicí jednotka D435, která tvoří nadřazený systém. Ta realizuje jednak víceosé řízení, ale i funkce pohybového kontroléru. Její použití je určeno pro náročné aplikace s dynamicky náročným polohováním. D435 spolu s S120 tvoří modulární celek jehož podobu zachycuje obr.

4.4. Signály z motorových enkodérů jsou přivedeny prostřednictvím tzv. Sensor Modulů (dále jen SM) do S120. Ta je složena z několika částí. Mezi ně patří CU320, Active Line module (dále jen ALM), Motor module (dále jen MM) a SM.



Obr. 4.4 Řídící jednotka Simotion D435 a frekvenční měnič Sinamics S120

Stejnoseměrný meziobvod frekvenčního měniče S120 je napájen z jednotky ALM, jehož stejnosměrné napájecí napětí se pohybuje na hodnotě 600V. ALM umožňuje vyrovnávat se s krátkodobými poklesy vstupního napětí a její přínosnou funkcí je možnost rekuperace energie zpět do napájecí soustavy sítě. S120 pracuje na výkonové úrovni řízení, zajišťuje řízení synchronních motorů v kaskádní regulaci (proud, rychlost, poloha). Stejnoseměrné napětí je z ALM přivedeno na svorky jednotlivých MM, které plní funkci střídačů na jejichž výstupech jsou IGBT tranzistory řízeny PWM modulací. Každý MM je připojen na pohonnou jednotku robota. Vyšší úrovně řízení jsou realizovány řídicí jednotkou D435, pro kterou byl výpočtový a řídicí algoritmus vytvořen.

4.3 Ovládací panel

Aby mohl operátor zadávat povel, je pro tyto účely určen mobilní ovládací

panel 177DP s dotykovým displejem (dále jen 177DP). Ten zajišťuje i pro programátora pohodlné ovládání zejména při ladění algoritmu, protože okamžitý zásah do toku celého algoritmu je pomocí definovaných obrazovek a není nutné hledat příslušné proměnné v Simotion Scout, jež svým vlivem působí na provádění algoritmu v čase. Jeho podobu zachycuje obr. 4.5. Panel je vybaven několika vnějšími klávesami, umístěnými po boku bezdotykové obrazovky. Ty slouží zejména pro předem naprogramované akce. U každé z nich jsou luminiscenční diody, které jsou také důležitým zobrazovacím prvkem. Na přední straně panelu se v horní části nachází bezpečnostní tlačítko TOTALSTOP, které slouží k okamžitému zastavení algoritmu a zablokování pohonů robota. Vedle bezpečnostního vypínače je otočný knoflík, který umožňuje zadávat inkrementální hodnoty polohy souřadnic v kartézských souřadnicích a definovat hodnoty úhlových natočení rotačních os. Ve spodní části panelu se nachází světelné tlačítko, které slouží k uvedení do provozu ALM a otočný zámek, pomocí něhož lze znemožnit celé ovládání panelu. Po levém a pravém boku panelu se v zadní části vyskytují dvě oválná tlačítka, která ovládají příkazy pro povolení resp. zablokování všech os robota při stisknutí, resp. při uvolnění. Ovládání robota pomocí tohoto panelu je snadnější než pomocí Simotion Scout a je doporučeno postupovat podle pokynů na jednotlivých obrazovkách.



Obr. 4.5 Ovládací panel

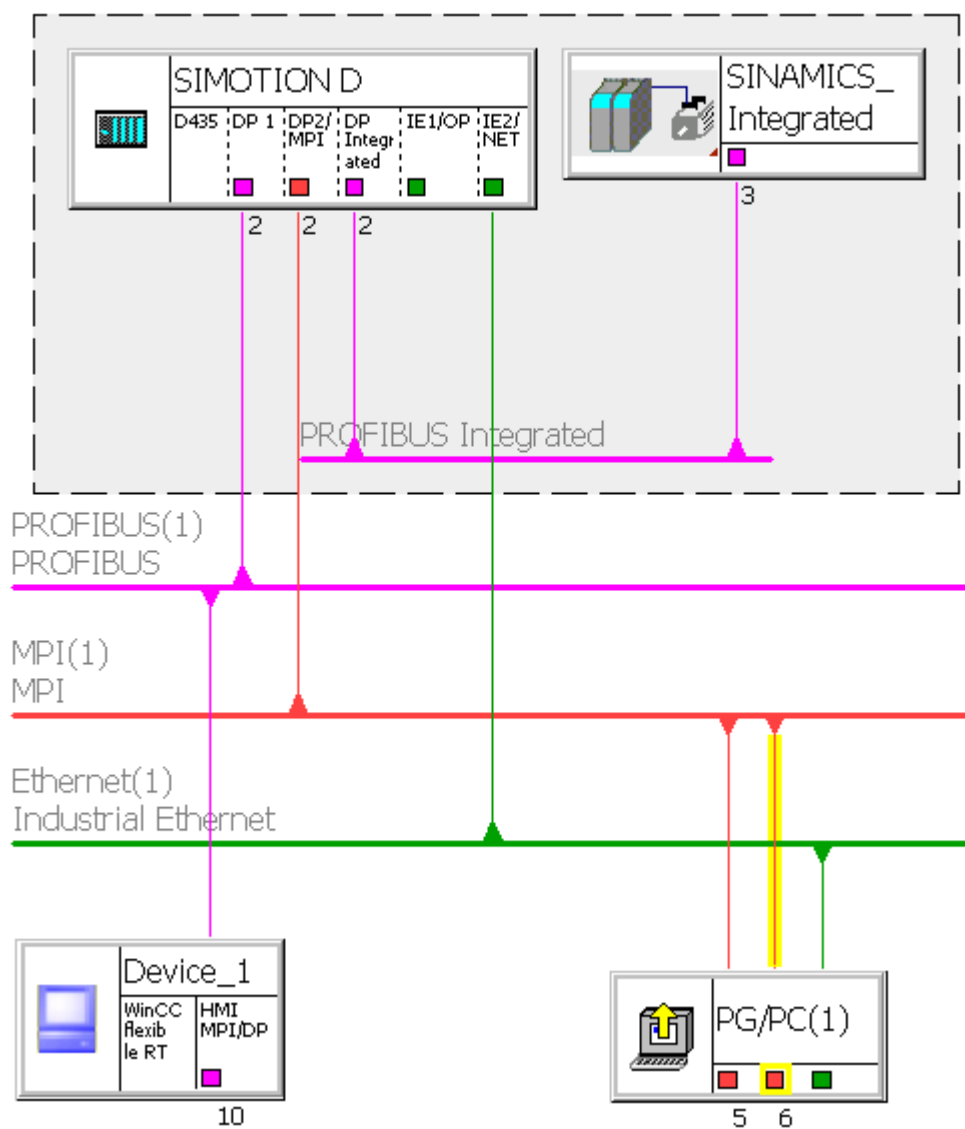
4.4 Vzájemná komunikace použitých HW zařízení

Mezi použitými hardwarovými prostředky, které byly v průběhu implementace algoritmů používány pro testování a odzkoušení bylo nutné definovat komunikaci. Způsob, jakým byly jednotlivé části vzájemně propojeny určovala daná topologie. Nastavení komunikace bylo provedeno v Simotion Scout pomocí nástroje *NetPro*. Jeho podobu znázorňuje obr. 4.6, na kterém je celá použitá hardwarová struktura.

Komunikace mezi D435 a 177DP a dále potom mezi D435 a S120 probíhala pomocí průmyslové komunikační sběrnici PROFIBUS-DP. Jedná se o nejjednodušší a nejrozšířenější variantu s rychlostí až 12 MB/s. Existuje totiž kromě PROFIBUS-DP, která je určena pro průmyslovou automatizaci také i PROFIBUS-PA, která se používá v oblasti procesní automatizace. Každá stanice, která je připojena pomocí sběrnice PROFIBUS-DP má svojí jedinečnou adresu v celé síti. Po této sběrnici docházelo k cyklické výměně informací mezi zmíněnými stanicemi. Jelikož D435 a S120 byla nakonfigurována jako zařízení typu master, jednalo se tedy o síť multimaster, na které může být několik masterů a podřízených jednotek (slaves).

Mezi PC, který byl určen pro programování řídicí jednotky D435 probíhala komunikace buď po síti Ethernet nebo pomocí sběrnice MPI. Jelikož nebyl PC vybaven speciální hardwarovou kartou, která by umožňovala komunikaci po MPI, bylo použito převodníku. Těch existuje celá řada, např. TCP/IP-MPI nebo USB-MPI. Použili jsme TCP/IP-MPI, který umožnil komunikaci po MPI s převodem na Ethernet. Tento typ komunikace byl nastaven zejména v případě, že bylo nutné programovat a přenášet aplikaci pro mobilní panel 177DP.

Jednotlivé modulární jednotky, z kterých je S120 sestavena a ostatní hardwarové části, které jsou přímo s ní spjaty (MM, SM, ALM) jsou vzájemně propojeny pomocí rozhraní DriveCliQ. Toto rychlá komunikační sběrnice je založena na technologii 100Mbs Ethernetu a má i stejný typ konektoru RJ-45. Urychluje a usnadňuje rozpoznávání připojených hardwarových částí jako motory, enkodéry apod. pro řídicí jednotku CU320. Její propojení s moduly S120 se musí přesně shodovat s topologií nastavenou v Simotion Scout.



Obr. 4.6 Prostředí *NetPro* pro definice komunikace mezi hardwarovými moduly

5. Realizace algoritmů

Jak již bylo zmíněno, navržená koncepce řídicího systému umožnila rozčlenit řízení do několika hierarchických úrovní. Tyto úrovně byly popsány v kapitole 3. Nejvyšší, tedy rozhodovací vrstva nebyla v diplomové práci uvažována, protože cílem bylo se zaměřit na tvorbu algoritmů, které se řadí do vrstvy transformační. Aby tato úroveň mohla být aplikována bylo nejdříve nutné vyřešit problematiku teorie přímé a inverzní úlohy robotiky, jejíž teoretická část byla vysvětlena v kapitole 2. Na základě konečného řešení bylo provedeno algoritmické ztvárnění matematických rovnic do programu MATLAB. To spočívalo v implementaci přímé a inverzní úlohy do dvou samostatných celků. Ověření správnosti výpočtů se provedlo dosazením výpočtů jedné úlohy do druhé a naopak. Pro snadnější představu byly znázorňovány pozice ramen robota pomocí uzlů a čar ve 3D grafech.

V další fázi se provedla konverze algoritmu z MATLAB do Simotion Scout, aby bylo umožněno zadávat požadované polohy a tím ovládat pohyby všech os robota. Poté, co jsme vyzkoušeli naprosto stejně navržený algoritmus na pohybech robota jsme obdrželi v určitých pozicích poněkud odlišné výpočty, než tomu bylo v MATLABu. Nejednalo se pozice robota, které by svým řešením znemožňovali zaujmout danou polohu, pouze se dopočítalo jiné vyhovující řešení. Cílem však bylo upravit navržený algoritmus, abychom výsledky dostali stejné. Tato nesrovnalost se týkala pouze inverzní úlohy, neboť je známo, že její řešení vychází nejednoznačné. U opačného typu úlohy, tj. u přímé je výsledkem pouze jediné řešení. Vyřešení obou matematických úloh nebylo součástí diplomové práce, cílem bylo provést její implementace.

Algoritmická podoba přímé a inverzní úlohy v MATLABu spočívala v interpretaci složitých výpočtů na základě vstupních hodnot typických pro konkrétní úlohu. Pro názornou představu jednotlivých řešení přímé a inverzní úlohy byla vypracována aplikace, jenž číselné výsledky znázorňovala i graficky. Po odladění algoritmu v MATLABU se zkonvertovaný program ověřil na robotovi. Kromě implementace přímé a inverzní úlohy se museli vytvořit funkce, které svým použitím ovládají jednotku S120. Tyto funkce zajišťují jednak výkonové ovládání jednotlivých servopohonů, ale i zpracování událostí na chyby, které mohou při provozu nastat.

Později byl algoritmus doplněn i o přímkové řízení pohybů koncového bodu a byla provedena parametrizace regulátorů pro jednotlivé regulační smyčky, aby bylo

dosaženo co nejlepší dynamické odezvy. V Simotion Scout nebylo možné graficky znázorňovat jednotlivá řešení daných algoritmů a funkcí, pouze jsme pomocí nástroje *Watch Table* mohli sledovat patřičné proměnné, které byly využity.

Každý z použitých softwarů počítal výsledky odlišně. Díky jiné přesnosti matematických výpočtů z obou vývojových prostředí vycházeli konečné hodnoty řešení úhlových natočení s určitým rozdílem. Tento rozdíl spočíval v číselném vyjádření ve stupních, které se lišilo řádově v tisícinách. Zpětné dosazení výsledků do opačného typu úlohy však ověřilo správné řešení v obou vývojových prostředích. Při zaokrouhlení na dvě desetinná místa bychom tyto hodnoty mohli vyhodnotit za identické. Zaokrouhlování jsme se vyhnuli, protože bychom zbytečně zanášeli do výpočtu chyby. Ač se může tato chyba na začátku zdát malá, v průběhu řešení se však nasčítá a v konečném výsledku je již zcela nezanedbatelná. Tím bychom sice jen nepatrně, ale určitě ovlivnili i koncové natočení rotačních členů robota. Ke konkrétním nepřesnostem a dalším nesrovnalostem, které výpočty ovlivňovali se vrátíme později v kapitole 5.6. Následující kapitoly popisují algoritmické podání obou typů úloh vysvětlené na příkladech v MATLABu.

5.1 Algoritmus přímé úlohy

Jak již bylo popsáno, přímá úloha zajišťuje transformaci mezi souřadnými systémy. V algoritmu se jedná o implementaci, která ze znalosti úhlových natočení jednotlivých rotačních členů řeší konkrétní orientaci a polohu v prostoru. Výsledkem je matice (1), jejíž tvar byl uveden v kapitole 2.1. Tato úloha umožňuje navíc kromě výpočtů zmíněné polohy a orientace také ověření výsledných řešení z úlohy inverzní.

5.1.1 Realizace v MATLABU

V přímé úloze zadáme jako vstupní parametry algoritmu úhlová natočení jednotlivých os do proměnných q_1 až q_6 ve stupních.. Dané rozměry ramen robota a vstupní hodnoty jsou potom dosazeny do koncové transformační matice, která vznikla vynásobením lokálních transformačních matic, jak bylo uvedeno v teoretické části 2.2.

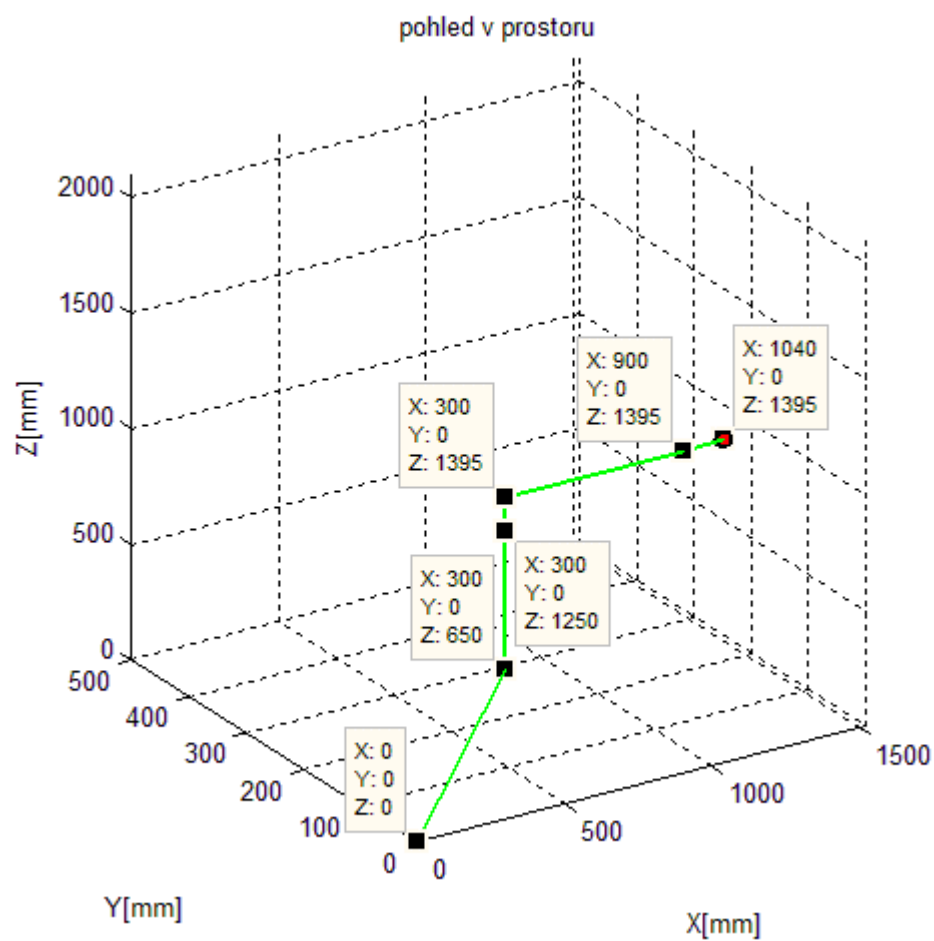
Výstupem algoritmu jsou hodnoty uspořádané v matici (1), která interpretuje polohu a orientaci koncového bodu robota vůči základnímu souřadnému systému.

Řešení přímé úlohy je jediné narozdíl od inverzní úlohy u které jich může být více.

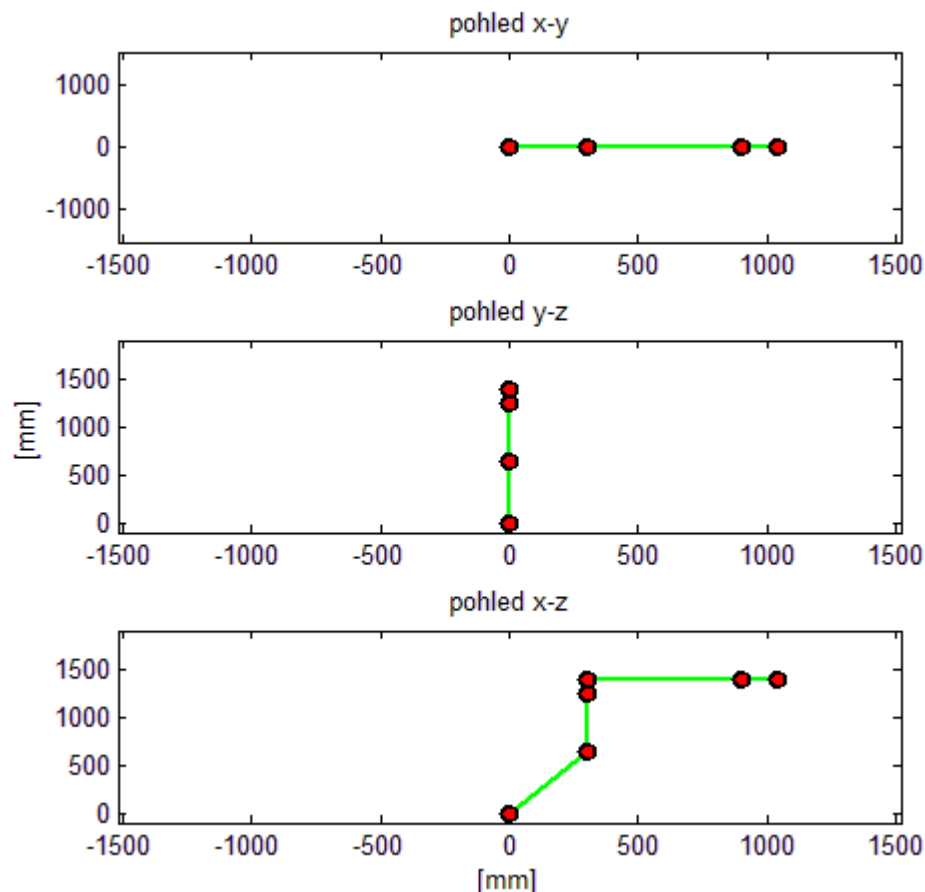
V základní poloze jsou souřadnice koncového bodu určeny hodnotami $P_x=1040$ mm, $P_y=0$ mm, $P_z=1395$ mm. Tato poloha může být dosažena několika různými hodnotami úhlových natočení, např. při nulových hodnotách stupňů na všech osách. S ohledem na základní polohu je však nutné dbát i na správnou orientaci. Tento fakt vychází totiž z inverzní úlohy, kdy je možné v základní poloze dosáhnout několika vyhovujících řešení úhlových natočení při určité orientaci. Např. při zadání nulových úhlových natočení ve stupních pro všechny osy dostaneme matici, která odpovídá tvaru matice (1). Výsledné hodnoty matice budou:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1040 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1395 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Pro lepší představu bylo využito graficko-početní znázornění o natočení jednotlivých os v prostoru formou uzlů a úseček ve 2D a 3D grafech. Aby bylo možné tyto grafy zakreslit počítají se transformační matice pro každý kloubový souřadný systém zvlášť. Polohu v základní pozici v prostoru zachycuje obr. 5.1. Bohužel z grafu nelze poznat o jakou orientaci koncového bodu se jedná. Lze ji však vyčíst z matice (2). Konkrétní představa o orientaci je patrná až na poloze koncového bodu na robotovi. Na obr. 5.1 jsou v jednotlivých uzlech zobrazeny i dané polohy, které v prostoru robot svými rotačními osy zaujímá. Tyto souřadnice jsou dány z rozměrů ramen robota a jsou udány v jednotkách milimetrů. Jednotlivé pohledy natočení ramen ve 2D zachycuje obr. 5.2. Na první pohled můžeme určit aktuální polohy kinematické struktury z nárysu, půdorysu a bokorysu



Obr. 5.1 Grafické znázornění natočení mechanismů robota v prostoru



Obr. 5.2 Grafické znázornění polohy mechanismů základní polohy v záporném smyslu orientace čelistí chapadla ve 2D

5.2 Algoritmus inverzní úlohy

Přirozenější a pohodlnější je zadávat souřadnice koncového bodu v kartézských souřadnicích. To znamená nalézt úhlové pozice rotačních členů ze znalosti polohy v kartézských souřadnicích. Tato úloha již není jednoduchá, neboť v drtivé většině případů se jedná o soustavu nelineárních rovnic. Narozdíl od přímé vnáší do řešení nejednoznačnosti. Existuje totiž více možností, jak cílovou polohu koncového bodu robota v prostoru zaujmout. Ta je dána úhlovým natočením jednotlivých os, které musí řešení vyhovovat. Pro každou pozici však toto platit nemusí a určitá poloha nemusí mít řešení žádné.

5.2.1 Realizace v MATLABU

V případě přímé úlohy byl algoritmus složen z poměrně jednoduchých

transformačních matic. Naproti tomu algoritmus inverzní úlohy je podstatně složitější a jeho výpočet i časově náročnější. Skládá se z rovnic, které byly získány z analytického řešení. Tyto rovnice mohou a nemusejí mít jednoznačná řešení, protože existuje více způsobů, jak se do cílové polohy dostat. Kromě reálných vznikají také imaginární. Těm je třeba se vyvarovat zejména v Simotion Scout, který s komplexními čísly neumí pracovat. Jsou to takové řešení, kterých robot o daných pevných rozměrech ramen není schopen dosáhnout. Imaginární řešení vznikají nerespektováním základních teorémů matematiky, jako že přepona je vždy delší než odvěsna, aj. Mimo jiné mohou při řešení inverzní úlohy vznikat i nedosažitelná řešení, které nerespektují polohové úhlové rozsahy jednotlivých os. Všechny výše zmíněné aspekty je nutné v algoritmech respektovat, aby výsledná řešení byla korektní. Konkrétní počet je pro většinu případů do od 0 do 4 řešení.

Vstupními hodnotami inverzní úlohy jsou vektory uspořádané do matice (1). Je na nás, jakou zadáme orientaci v jednotkových vektorech a polohu v kartézských souřadnicích koncového bodu. Je však nutné mít na paměti, že matice orientace musí být zadána, tak aby ve všech sloupcích a řádcích byla hodnota vždy jen na jednom místě. Celková matice orientace musí být ortonormální. Orientaci zadáváme pomocí druhého a třetího vektoru. První vektor je počítán z těchto dvou kartézským součinem, tak aby soustava byla vzájemně kolmá a pravotočivá.

Algoritmus dosazuje do získaných rovnic hodnoty celkové vstupní matice a počítá úhlová natočení jednotlivých kinematických členů. První úhlové natočení kinematického členu je získáno z hodnot matice a z rozměrů ramen robota. Následující výpočty kloubových proměnných jsou založeny jednak na vstupní matici a z hodnot rozměrů ramen, ale i z úhlových pozic předešlých členů. Nepřesnost výpočtu předcházejícího úhlu může ovlivnit přesnost výpočtu následujícího. Tímto se pak tato nepřesnost kumuluje. Poté téměř zanedbatelná nepřesnost při výpočtu prvního úhlu může hrát v konečném výsledku velikou roli. Právě i tento problém jsme museli řešit při výpočtech dvěma různými výpočetními prostředky MATLAB a D435, který bude detailně popsán v kapitole 5.6 V algoritmu jsou implementovány potřebné rozhodovací podmínky, které slouží v případě, aby byla zaručena správnost běhu algoritmu a výsledné řešení bylo v reálném oboru čísel. Pokud má úloha jednoznačné řešení, tzn. existují definované úhly rotačních os může koncový bod zaujmout žádanou polohu. Tu lze vybrat z konečného počtu správných řešení inverzní úlohy.

Výstupem algoritmu jsou uspořádané vektory o rozměrech 1x6, které udávají natočení ve stupních všech reálných rotačních členů kinematické struktury. Celkový počet konečných možných řešení je variabilní. Je ovlivněn mnoha aspekty, které byly výše zdůvodněny.

Díky symetričnosti stejné osy, která je společná pro 4 a 6 osu robota, je možné dosáhnout nekonečného množství řešení, ale jen v případě, že pátá osa má nulové natočení, tedy že jedna z os obou kloubových souřadných systémů je totožná.

Pro názornou ukázkou si uvedeme příklad inverzní úlohy. Zadáme-li základní polohu v kartézských souřadnicích s orientací koncového bodu v kladném x-směru bude vstupní matice ve tvaru určenou vztahem (3). Vztah (3) je podobný (2), ale s jinou orientací chapadla!

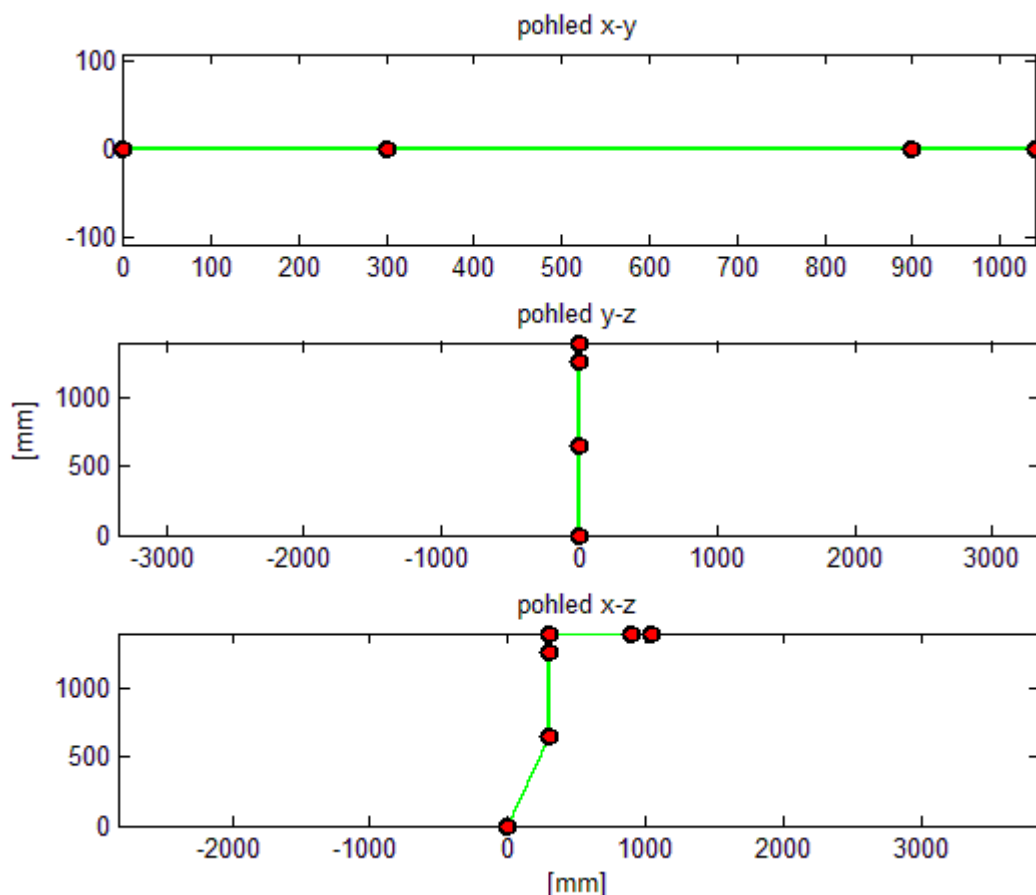
$$A = \begin{bmatrix} 0 & 0 & 1 & 1040 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1395 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Vektor orientace čelistí chapadla je v tomto případě ve kladném y-směru. První vektor byl dopočítán, jak bylo zmíněno výše. Výstupem jsou řešení, které udávají úhlová natočení všech os robota ve stupních. V tomto případě jsou k dispozici čtyři řešení ve tvaru.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 78,69 & 77,69 \\ 0 & 0 & -152,83 & -152,83 \\ 0 & 180 & -180 & 0 \\ 0 & 0 & -75,13 & 75,13 \\ -180 & 0 & 0 & -180 \end{bmatrix}$$

Jak již bylo zmíněno o společně procházející ose u 4 a 6 souřadného systému při nulovém natočení 5 osy bychom mohli uvažovat o nekonečném počtu řešení. Právě v základní pozici se projevila tato skutečnost a výsledná řešení z MATLABu a D435 nebyla totožná. Byla zachována jak poloha, tak i orientace. Drobnou modifikací algoritmu jsme tento vzájemný rozdíl minimalizovali. Řešením tohoto problému se

zabývá samostatná kapitola 5.6. Stejně jako v přímé úloze byla i algoritmická část inverzní úlohy v MATLABu doplněna grafickým znázorněním o poloze jednotlivých kinematických členů. Grafické znázornění výpočtů z prvního sloupce podle inverzní úlohy je na obr. 5.3.



Obr. 5.3 Grafické znázornění polohy mechanismů základní polohy v kladném smyslu čelistí chapadla ve 2D

5.3 Řízení po přímce

Nezbytnou součástí běžných řídicích systémů průmyslových robotů bývají interpolační nástroje, jejichž algoritmy umožňují vykonávat pohyby po prostorové křivce. Jedná se např. o lineární, kruhové, či spline-interpolace, které mimo jiné podporují i pohony S120. Tyto interpolace však museli být pro 6-osý systém vyvinuty, neboť jednotka umožňuje pouze interpolace pro 4-osé systémy, jak již bylo zmíněno. Konkrétní pohyby koncového bodu robota jsou určeny podle technologického postupu v daném průmyslovém odvětví. Díky této možnosti byl vytvořen výpočtový algoritmus, který na základě inverzní úlohy vypočítal soubor hodnot úhlových natočení jednotlivých

os a po jeho ověření v MATLABu byl implementován také do Simotion Scout. Algoritmická část je založena na vztahu (4), který podle definice počátečního a koncového bodu provede přímkovou interpolaci.

$$X^i = A + i \frac{\mathbf{P}}{n} \quad (4)$$

A určuje počáteční bod, který je zadán ve třech souřadnicích v kartézském souřadném systému. Vektor P je určen rozdílem koncového a počátečního bodu, tj. $P=(B-A)$, ve všech třech souřadnicích. Proměnná i obsahuje hodnotu iteračního kroku a celkový počet těchto kroků je určen počtem interpolačních bodů proměnné n . Výsledek je uložen v každém iteračním kroku do proměnné X , která třemi souřadnicemi vstupuje do inverzní úlohy. Z ní je potom vypočítáno potřebné úhlové natočení všech rotačních os. Ty vstupují dále u každé osy do polohových profilů, kde podle lineární interpolace je vykonán příslušný pohyb.

5.4 Testování algoritmů

Po implementaci do prostředí MATLAB byla provedena ověření výpočtů z přímé a inverzní úlohy. Testování spočívalo dosazování výsledků z jedné úlohy do druhé a naopak. Tuto část jsme potom provedli i v Simotion Scout, kde jsme sledovali jednak výstupní hodnoty algoritmů, ale i konkrétní pozice robota. Ověření výpočtů při řízení po přímkové trajektorii bylo ověřeno taktéž. Více se měřením zabývá samostatná kapitola 5.7

5.5 Konverze algoritmu do prostředí Simotion Scout

Otestovaný a odladěný algoritmus ve vývojovém prostředí MATLAB byl později zkonvertován do Simotion Scout. Nejprve se jednalo o algoritmické ztvárnění přímé a inverzní úlohy robotiky, později byl přidán i program umožňující pohyb po přímkové trajektorii, viz dále. Pro správnou funkci všech hardwarových zařízení však museli být definovány i bezpečnostní funkce zaručující obsluhu událostí na chyby zařízení, dále příkazy pro ovládaní pohonů a v neposlední řadě i konverzní funkce pro převody veličin. Simotion Scout umožňuje nastavovat různé parametry hardware, limity rychlostí, limity polohy, dynamiku pohonů a mnoho dalších funkcí pomocí obslužných

průvodců, jejichž popis by byl nad rámcem této práce. O možností Simotion Scout se lze dozvědět v [8].

Syntaxe jazyka v tomto vývojovém prostředí je odlišná vůči strukturám v MATLABu. Jedná se o jazyk ST (Structure language), který má podobnou syntaktickou skladbu jako Pascal. V Simotion Scout jsou patřičné funkce, procedury, programy či podprogramy součástí ST unit, které jsou cyklicky vykonávány v D435. Tyto ST unity vlastně obsahují příslušný algoritmus, který je členěn do menších algoritmických celků z důvodu přehlednosti a snadnějších zásahů do celého algoritmu. Použity jsou ST unity:

- *A_cycle*

Který obsahuje podprogram *start_UP* ve kterém se nastaví softwarové limity úhlových natočení os, aktuální rychlosti, a některé potřebné proměnné. Tato část se vždy vykoná po uvedení D435 z vypnutého do zapnutého stavu. Dále je obsažen hlavní program *cycle*, který je páteří programu, z kterého se volají všechny funkce. Poslední podprogram má název *safeposition* a slouží k uchování aktuálních poloh do tzv. retain variables (trvalých proměnných), které své hodnoty neztratí, ani při odpojení od elektrické sítě. Ten je vždy vykonán při změně D435 ze zapnutého do vypnutého stavu.

- *Global_variables*

Tento program slouží pro deklaraci datových typů a všech použitých globálních proměnných a konstant, které jsou využívány ve všech programech.

- *Interpolation*

Zde jsou definovány algoritmické celky, umožňující pohyb po přímkové trajektorii. Jedná se o polohové profily všech os, které později vykonávají pohyb po spline křivkách a výsledkem je potom přímkový pohyb koncového bodu.

- *My_function*

Obsahuje funkce, které si jednotlivé programy volají. Z důvodu přehlednosti a snadné úpravy bylo přistoupeno na tvorbu funkcí do samostatného programu. Obsaženy jsou funkce pro ovládání pohybů robota, tak i bezpečnostní funkce sloužící k okamžitému zastavení.

- *Transformations*

Výpočtová část přímé a inverzní úlohy a k ní i další potřebné funkce jsou

implementovány v této části programu.

Podobně jako v MATLABu i zde má výpočtová část algoritmu svojí úlohu, která se dělí na přímou a inverzní. Dále se v celém projektu vyskytují programové bloky a funkce, které jsou pro řízení pohybů robota nutné, jak bylo výše zmíněno. Ukázkou řešení přímé a inverzní úlohy nemusíme uvádět, protože je více názorná z předchozí kapitoly. V Simotion Scout lze proměnné číst, zapisovat a sledovat. Jednotka D435 provádí cyklické vykonávání všech ST unit a jednoduše krokovat v Simotion Scout nelze. Při krokování se totiž cyklus přeruší a dojde i odpojení napájení ALM, což znemožní pohyby robota. To byl také důvod proč byl nejdříve odladěn algoritmus v MATLABU. Prostředí Simotion nabízí jednoduché grafické nástroje, pomocí kterých lze sledovat hodnoty proměnných v čase. Nedají se však vytvářet grafické výstupy závislé veličiny na nezávislé jako v MATLABu, který je pro tyto účely vybaven. Běh toku algoritmu je možný měnit v Simotion Scout, ve většině případů bylo zvoleno ovládání pomocí dotykového mobilního panelu, který zejména zajistil pohodlné ovládání robota.

Pohyb po přímkové trajektorii byl vypočítán pomocí interpolačního vzorce 4.0, který spočítal body v kartézské souřadném prostoru. Výpočty v MATLABu i Simotion Scoutu vycházeli totožně, proto se přistoupilo na definice polohových profilů jednotlivých os, které tento software nabízí ve svém širokém spektru funkcí. Tyto profily nabízejí mnoho interpolačních metod a slouží k vytváření pohybů rotačních os v čase podle těchto křivek. Jedná např. se o lineární, Bezierovu či kubickou interpolaci. Na základě výpočtů z inverzní úlohy je do každého profilu osy uložena hodnota úhlového natočení a poté je provedena interpolace mezi těmito body. Jakmile jsou všechny profily vytvořeny a spuštěny v jeden okamžik, jsou pohyby všech rotačních os vhodně sesynchronizovány a koncový bod koná přímkový pohyb.

5.6 Chyby a nepřesnosti algoritmu

Po implementaci algoritmu do Simotion Scout bylo nutné zajistit, aby výsledky byly totožné s výsledky z MATLABu. Inverzní i přímá úloha počítá v plovoucí řádové čárce a každý výpočetní software interpretuje výsledky s jinou přesností, což v konečné instanci mělo za následek jiné hodnoty. U přímé úlohy se tento problém nevyskytl a tak bylo předmětem řešit algoritmus úlohy opačné. Jiná řešení inverzní úlohy byly

vyhodnoceny jako správné, protože byla dosažena požadovaná cílová poloha koncového bodu i jeho orientace, pouze s jiným úhlovým natočením os robota. Pokusme se nastínit v čem konkrétní problém spočíval a jaké bylo konečné řešení.

Při výpočtech úhlových natočení jednotlivých os jsou volány inverzní funkce goniometrických funkcí. Vstupem jsou hodnoty dané podílem odvěsny ku přeponě. Aby bylo zaručeno řešení v oboru reálných čísel musí být přepona větší než odvěsna trojúhelníka. Pokud je splněna tato podmínka, je podle hodnoty podílu spočten patřičný úhel v oboru hodnot $H_f \in \langle 0, 2\pi \rangle$. Právě jiné hodnoty odvěsny a přepony byly příčinou odlišných úhlových natočení. Pro konkrétní představu v MATLABu vycházela v základní poloze robota hodnota první odvěsny $H_x = -5,9016 \cdot 10^{-8}$, druhá odvěsna $H_y = 0$, přepona $H = 5,9016 \cdot 10^{-8}$. V Simotion Scout $H_x = 6,1232 \cdot 10^{-17}$, $H_y = 0$, $H = 6,1232 \cdot 10^{-17}$. Obě hodnoty bychom mohli položit téměř za nulové, to jsme zpočátku také provedli, ale dopustili jsme se tímto chyby, která ovlivnila následující výpočty. Díky řetězení chyb v průběhu výpočtu jak bylo popsáno v části 5.2.1 docházelo ke značným chybám. V některých případech došlo i k vyřazení řešení z výpočtového algoritmu díky podmínkám. Zaokrouhlovat mezivýsledky se tedy ukázalo jako nevyhovující. Abychom byly ještě více konkrétní, úhlové natočení se neshodovali pouze na 4 a 6 ose, které mají společnou procházející osu, což je dáno z konstrukce ramene robota. Díky tomu, že obě osy mají omezení v rozmezí $\langle -\pi, \pi \rangle$ mohli jsme provést změnu u 4 osy, která spočívala v přiřazování hodnoty přepony do odvěsny, za předpokladu, že absolutní hodnoty přepony a odvěsny byly totožné. U této osy je počítán úhel pomocí inverzní funkce \arccos . Jakmile jedna z hodnot měla opačné znaménko (např. došlo k přepsání záporné hodnoty kladnou), přičemž absolutní hodnoty byly vzájemně totožné, chyby jsme se nedopustili. Díky tomu, že 4 osa může zaujmou polohu v rozmezí $\langle -\pi, \pi \rangle$, je po přiřazení k původní hodnotě odečteno π (v případě přepsání kladné hodnoty zápornou je naopak přičteno π). Dále se 6 osa přesně natočí tak, aby byla zachována orientace koncového bodu. Tím pádem je možné jakkoliv 4 osu úhlově natočit a k ní provést natočení 6 osy, aby řešení zůstalo zachováno. To je možné pouze při nulovém natočení 5 osy. V jiných polohách toto možné není. Právě tyto neshodnosti se týkaly pouze základní polohy mechanismů robota tj. $(x, y, z) = (1040, 0, 1395)$, při které je úhlové natočení 5 osy nulové. V ostatních polohách byly výsledky v MATLABu a Simotion Scoutu totožné.

Další chybu, kterou jsme obtížně dohledali byla nedostatečná definice Ludolfova

čísla v Simotion Scout. To se stejně jako v předchozím případě týkalo inverzní úlohy. Zatímco v MATLABu je tato konstanta implementována, v Simotion Scout musí být definována dodatečně. Nejprve byla deklarována s rozlišením na 2 desetinná místa, což se ukázalo jako nedostatečné, protože se jednak výsledky rapidně lišili, ale v některých případech bylo zaznamenáno i méně vyhovujících řešení. Nakonec se konstanta definovala jak v MATLABu, tak i v druhém vývojovém prostředí a to na stejný počet desetinných míst, konkrétně 120. Tím byla zaručena dostatečná přesnost mezivýpočtů a následně i konečných řešení.

Po těchto úpravách se výsledky podařilo ztotožnit. Vzájemný rozdíl hodnot konečných výsledků se pohybuje v řádů tisíci stupňů. Při zaokrouhlení přibližně na dvě desetinná místa dosáhneme identity. Zaokrouhlením však nepatrně změníme polohy úhlových natočení a můžeme tím i ovlivnit polohu koncového bodu. To je tématem další kapitoly, která se týkala ověřování výpočtů pomocí improvizovaných měřících metod.

5.7 Měřené hodnoty

Aby bylo možné ověřit správnost výsledků v Simotion Scout na pohybech robota, byl proveden soubor měření. V první fázi se hlavně jednalo o prověření inverzní úlohy, potom se testovala správnost výpočtů přímé úlohy. Na konci měření se hodnoty porovnali vzájemně z obou vývojových prostředí.

Nejprve se zadali se žádané hodnoty polohy koncového bodu robota v kartézských souřadnicích a porovnávaly se s hodnotami, které byly naměřeny pomocí experimentálních měřících metod. Určili jsme přibližně počátek souřadného systému pro pracovní prostor robota a odměřovali jsme polohy koncového bodu pomocí jednoduchého měřidla. Vzhledem k tomu, že robot je připevněn k mobilnímu rámu, který znemožnil přístup k referenčním souřadnicím, bylo obtížné najít střed souřadného systému. Improvizovanou metodou jsme zjistili přibližné souřadnice v prostoru a porovnali je s žádanými hodnotami, které vstupovali do inverzní úlohy. Tyto měření byly pro nás pouze orientační, proto také nejsou v diplomové práci uvedeny. Tímto jsme tedy ověřili inverzní úlohu a došli k závěru, že polohy jsou se shodují.

Přímá úloha byla ověřována pomocí výpočtů z inverzní úlohy. Výsledky museli být dosazeny do přímé úlohy nezaokrouhlené, protože zpětné řešení by po zaokrouhlení

bylo méně přesné. Výpočty by se však lišili řádově v tisícinách. Stejně tomu bylo i naopak, tj. výpočty z inverzní úlohy následně dosazené do přímé. Algoritmus je tedy citlivý na zaokrouhlování u obou typů úloh. V příloze 1 jsou naměřené hodnoty inverzní úlohy v základní poloze při všech možných orientacích, které je možné zadat. Ověřovali jsme polohu, orientaci a pomocí experimentální metody i skutečnou polohu v pracovním prostoru robota. Nakonec jsme vše porovnali s výsledky v MATLABu. Tato měření byla provedena s modifikovaným algoritmem, jehož podstata byla popsána v předcházející kapitole.

Byla zjištěna neshodnost výpočtů při dvou zadaných orientacích v základní poloze. Robot při výpočtu inverzní úlohy koncovou polohu s příslušnou orientací sice zaujal, ale po zpětném dosazení úhlových natočení do přímé úlohy jsme matici orientací dostali v jiném tvaru. V příloze 1 je v tabulce vyznačen sloupec M a P, který obsahuje znaky A (ANO), N (NE). M sloupec značí vzájemnou shodnost výpočtů mezi MATLAB a D435 a sloupec P vyjadřuje, zda po dosazení výpočtů z inverzní úlohy do přímé bylo dosaženo totožných výpočtů. Z hlediska širokého rozsahu celé práce se z časových důvodů tyto chyby nepodařilo v době, kdy byl tento text psán odstranit.

6. Nastavení regulátorů

Systém pohonů SINAMICS, který byl určen pro řízení os robota obsahuje širokou sadu funkcí, mezi něž patří různé typy regulačních struktur, řízení pohybů podle profilů, nastavení dynamiky os, či nástroje určené pro konfiguraci samotných motorů. Toto široké spektrum funkcí je k dispozici v software Simotion Scout, který je určen pro konfigurace D435, tak i S120. Jelikož průmyslový robot je potřeba ovládat s danou přesností, je nutné nastavit optimální dynamiku všech pohonů, aby byla zaručena co nejlepší odezva na žádanou hodnotu. Tyto parametry umožňuje Simotion Scout měnit a tím ovlivňovat nastavení modulárního systému S120, který ovládá jednotlivé pohony robota. Na základě této možnosti byla diplomová práce doplněna i o část, která se zabývá návrhem jednotlivých parametrů regulačních smyček. Díky tomu bylo možné zlepšit dynamické vlastnosti všech rotačních os a byla zaručena odpovídající odezva na žádanou polohu v kratším čase, než tomu bylo při výchozích parametrech. To bylo potřeba zvláště při přímkovém řízení, kdy jsme potřebovali provést pohyb po trajektorii co nejdynamičtěji. Při něm dochází u jednotlivých os k velkým změnám polohy v krátkém čase. Před popisem nastavení optimálních parametrů regulátorů se zaměříme nejdříve na některé základní pojmy, které se v regulaci používají.

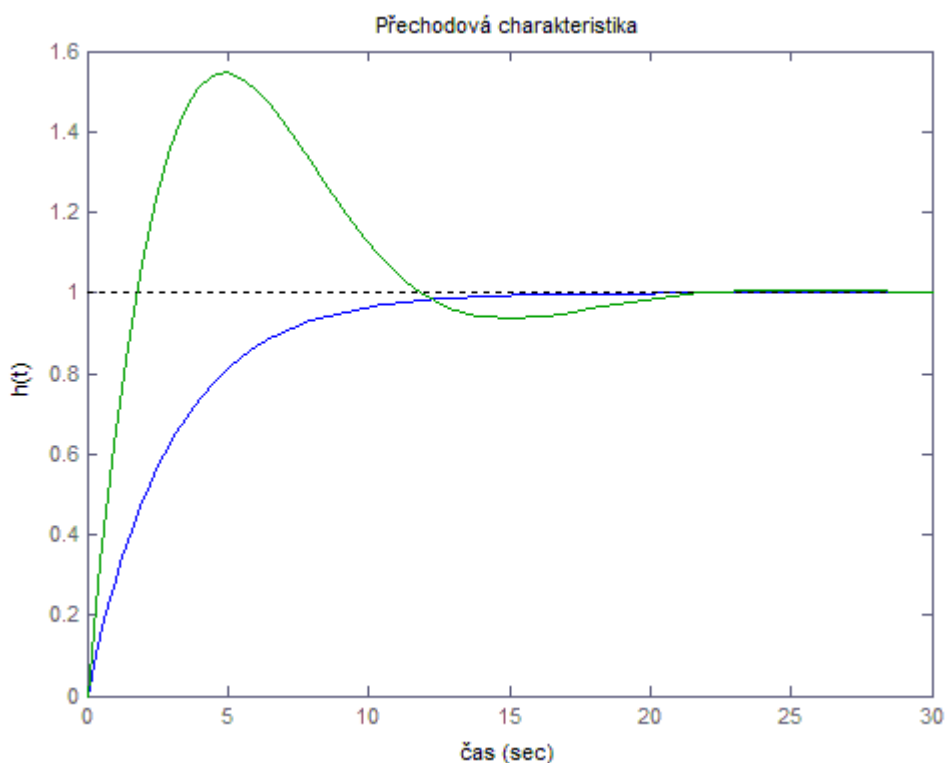
6.1 Teorie regulace

Regulací rozumíme udržování určité technologické veličiny (tzv. regulované veličiny) na určité hodnotě (tzv. žádané hodnotě). Tuto funkci realizuje regulátor, pokud se jedná o automatickou regulaci. Jako regulovanou soustavu chápeme zařízení na kterém je regulace prováděna. Na základě těchto dvou pojmů vzniká uzavřený regulační obvod se zpětnou vazbou, který umožňuje tuto činnost provádět.

6.1.1 Typy regulátorů

Obecně regulátory můžou vstupní regulační odchylku zesilovat (P), integrovat (I) či derivovat (D). Tyto jednotlivé parametry svým synergickým působením podmiňují průběh celého regulačního pochodu. Správné nastavení těchto konstant se potom promítne do ovlivnění výstupní akční veličiny. Optimální nastavení regulátoru a soustavy je možné charakterizovat pomocí přechodové charakteristiky. Nejčastěji bývá vstupním signálem jednotkový skok a na výstupu se měří přechodový děj. Kvalita

regulace se potom posuzuje překmity, zákmity či dobou ustálení regulované veličiny. Přechodové charakteristiky jsou uvedeny na obr. 6.1, který znázorňuje chování dvou různých soustav. Optimální nastavení parametrů regulátoru dokazuje průběh, který je zakreslen modrou barvou. Druhý průběh zakreslený barvou zelenou již vykazuje jiné nastavení těchto parametrů. Záměrně není uveden matematický popis regulované soustavy, neboť není předmětem se zabývat detaily. Více se této problematice dozvědět v literatuře [7].



Obr. 6.1 Přechodová charakteristika dvou dynamických systémů

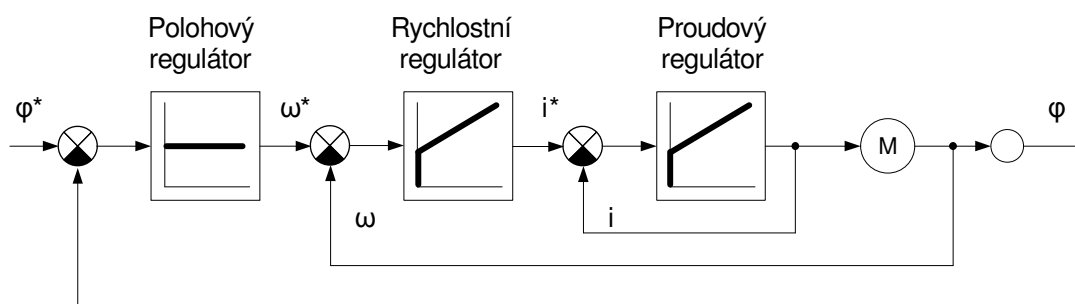
6.1.2 Zásady regulace

Aby regulační pochod netrval dlouho a brzy se ustálil a aby překmit regulované veličiny nebyl příliš velký je nutné nastavit optimální parametry regulátorů. Existuje mnoho různých metod pro nastavení konstant regulátorů, např. metoda minima kvadratické plochy, Ziegler-Nicholsova či Cianconeho korelace, aj. V praxi je v oblasti servopohonů využívána metoda Ziegler-Nicholsova (dále jen Z-N), která je dosti rozšířená a jednoduchá a je založena na seřízení, které přivede regulační obvod na hranici stability. Proporcionální konstantu zvyšujeme, tak aby se obvod rozkmital,

přičemž integrační konstanta je zvolena maximální a derivační úplně vyřazena. Při těchto kmittech stačí zjistit časovou periodu kmitů na výstupu a společně s proporcionální konstantou dosadit do empirických vztahů pro zvolený typ regulátoru. Podrobnosti o nastavení parametrů regulátorů jsou uveřejněny např. v [16].

6.1.3 Kaskádní regulace

Častou aplikací kaskádního regulačního obvodu je při řízení veličin s rozdílnou dynamikou, nebo při řízení motorů. U pohonů se využívá kaskádní struktura podle obr. 6.2, která ve své kaskádě používá 3 zpětné vazby. Na vstupu je žádaná hodnota úhlového natočení a do rozdílového členu vstupuje aktuální hodnota úhlového natočení. Za ním je regulační odchylka přivedena do polohového regulátoru typu P, z kterého vystupuje žádaná rychlost otáček. Ta dále vstupuje do rozdílového členu, kde je odečtena od tohoto signálu aktuální rychlost. Regulační odchylka potom směřuje do rychlostního regulátoru typu PI a z něj již vystupuje řídicí veličina proudu. Hodnota aktuálního proudu se odečte od žádané hodnoty proudu a výsledná hodnota vstupuje do proudového regulátoru typu PI. Na jeho výstupu je regulovaná veličina napětí.



Obr. 6.2 Blokové schéma kaskádního regulačního obvodu

Proudová smyčka by měla být ze všech co nejrychlejší a její frekvenční pásmo co nejširší, aby nedošlo ke zpomalení či frekvenčnímu útlumu regulované veličiny proudu. Žádané veličiny proudu jsou generovány z nadřazené rychlostní smyčky a požadované hodnoty rychlosti potom z polohového regulátoru. Rychlostní, nebo-li otáčková smyčka musí regulovat rychlost při určité změně momentu a polohová smyčka má za cíl regulovat podle řídicí hodnoty úhlového natočení.

6.2 Nastavení parametrů regulátorů

Jednotka SINAMICS dovoluje nastavovat parametry regulátorů pomocí grafických nástrojů v prostředí Simotion Scout, jak již bylo zmíněno. Dovoluje aplikace jednoduchých kaskádních struktur i použití složitějších regulačních struktur např. dopředné řízení, adaptivní regulace aj. Tato bohatá nabídka umožňuje řízení od jednoduchých servopohonů až po náročné aplikace ve výrobních strojích. S120 používá k regulaci pohonů průmyslového robota kaskádní regulační struktury, u kterých lze nastavovat patřičné konstanty v jednotlivých regulátorech pomocí průvodců v Simotion Scout a tím lze ovlivňovat celkový regulační pochod. Základní schéma, které bylo uvedeno v teoretické části je využíváno i pro regulaci pohonů průmyslového robota. V pohonech SINAMICS je však podstatně rozmanitější, neboť dovoluje nastavovat více parametrů, které regulaci může ovlivnit jako například dopředné regulace, filtry žádaných hodnot aj.

Žádanou hodnotou je jednak poloha a rychlost, ale i proud. Signál o poloze je měřen 3-pólovými resolvers u každého pohonu a je přiveden pomocí SM do S120. Aktuální rychlost vstupuje do proudového regulátoru z Motor Encoderu, který převádí matematickou funkcí aktuální polohu danou resolverovým odměřováním. Výstupem proudového regulátoru je regulovaná veličina napětí, podle které jsou vhodným způsobem řízeny jednotkou CU320 IGBT tranzistory jednotky MM.

Jak již bylo v teorii zmíněno proudová zpětná vazba je nejnvtířnější a také navržena nejrychlejší a stanovení konkrétních parametrů začíná právě od této zpětné vazby. Stejným způsobem bylo postupováno i při návrhu konstant jednotlivých regulátorů pohonů S120 v Simotion Scout. Při měření a nastavování konstant regulátorů byly všechny motorové pohony na výstupu zatíženy vlastními hmotami robota, proto bylo obtížné stanovit vhodné konstanty, které by vyhovovali ve všech pracovních polohách. Jelikož se v různých polohách měnili momenty setrvačnosti na příslušných osách, bylo by žádoucí řízení doplnit o kompenzaci tohoto vlivu. Konečné nastavení parametrů však bylo vyhovující oproti původním.

Postup návrhu konstant spočíval nejdříve ve využití automatických identifikačních funkcí, které zajistili dobré nastavení konstant proudových, rychlostních a polohových regulátorů. Při této funkci jsou nejprve změřeny mechanické parametry a potom je proveden návrh parametrů proudového a nakonec rychlostního regulátoru.

Jelikož se nejedná o jednohmotový systém, jsou naměřené charakteristiky zatížené faktory, které představují různé rezonanční špičky, které jsou patrné z naměřených charakteristik. Na základě této skutečnosti jsou k dispozici v S120 v dané smyčce regulační struktury bloky pro přídavné přenosové funkce, které představují filtry typu dolní propust, pásmová propust, aj., které lze nastavovat pomocí Simotion Scout

V dalším kroku jsme provedli experimentální přenastavení proporcionálních a integračních složek proudového a následně rychlostního regulátoru K doladění regulační soustavy jsme využili jen modifikované metody Z-N. K tomu bylo samozřejmě nutné naměřit vždy přechodové a frekvenční charakteristiky a provést jejich vyhodnocení. Je třeba říci, že u některých os se experimentálně nastavené parametry od parametrů nastavených automatickou ladící funkcí moc nelišili. Avšak vhodnou změnou jsme zlepšení dynamiky pohonů dosáhli. Pokud bychom neměli k dispozici automatické identifikační funkce, mohli bychom konstanty navrhnout podle některých známých metod jako jsou např. zmíněná Z-N, metoda optimálního modulu či metoda symetrického optima. Podrobněji jsou tyto metody popsány v [10].

V poslední fázi jsme použili jako v předchozím případě automatické ladící procedury pro nastavení proporcionálního regulátoru v polohové smyčce. Vypočtené konstanty nebylo potřeba experimentálně přenastavovat, protože orientační měření frekvenční a přechodové odezvy se na změnu od nominálně vypočtené hodnoty příliš nelišilo. Zaměřili jsme se hlavně na identifikaci přechodové a frekvenční charakteristiky pro proudovou a otáčkovou zpětnou vazbu, které by měli být rychlejší než polohová smyčka. Naměřené charakteristiky jsou dále.

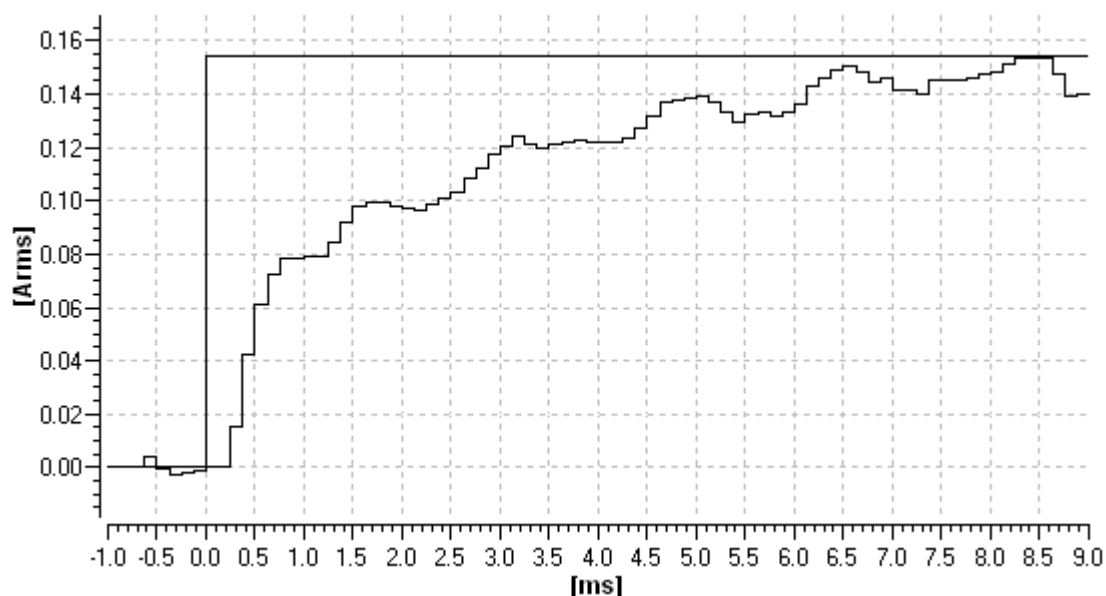
V tab. 1 jsou uvedeny konstanty regulátorů proudové, rychlostní a polohové smyčky, které byly jednotkou S120 zvoleny jako výchozí. Tab. 2 obsahuje námi nastavené parametry, tj. změřené automatickými identifikačními funkcemi a následně přenastavené pro proudovou a otáčkovou vazbu. Pro polohovou smyčku byly konstanty ponechány na hodnotách změřených automatickou identifikační funkcí, jak bylo zmíněno. Měření charakteristik bylo provedeno na všech šesti osách robota, které spočívalo v měření zatížených motorů, což mohlo ovlivnit výsledné charakteristiky. Pro názornost jsou uvedeny naměřené průběhy pro 4 osu. Jedná se o amplitudové, fázové a přechodové charakteristiky pro proudovou a rychlostní smyčku. Na obr. 6.3 a 6.4 je znázorněna přechodová charakteristika proudové smyčky před a po nastavení konstant regulátorů.

číslo osy	regulační smyčka				
	proudová		rychlostní		polohová
	P [V/A]	Ti [ms]	P [Nm.s/rad]	Ti [ms]	P [1/s]
1	9,569	2	1,248	10	10
2	4,389	2	0,855	10	10
3	8,428	2	0,593	25,16	10
4	17,570	2	0,028	50	10
5	17,748	2	0,017	50	10
6	15,955	2	0,024	50	10

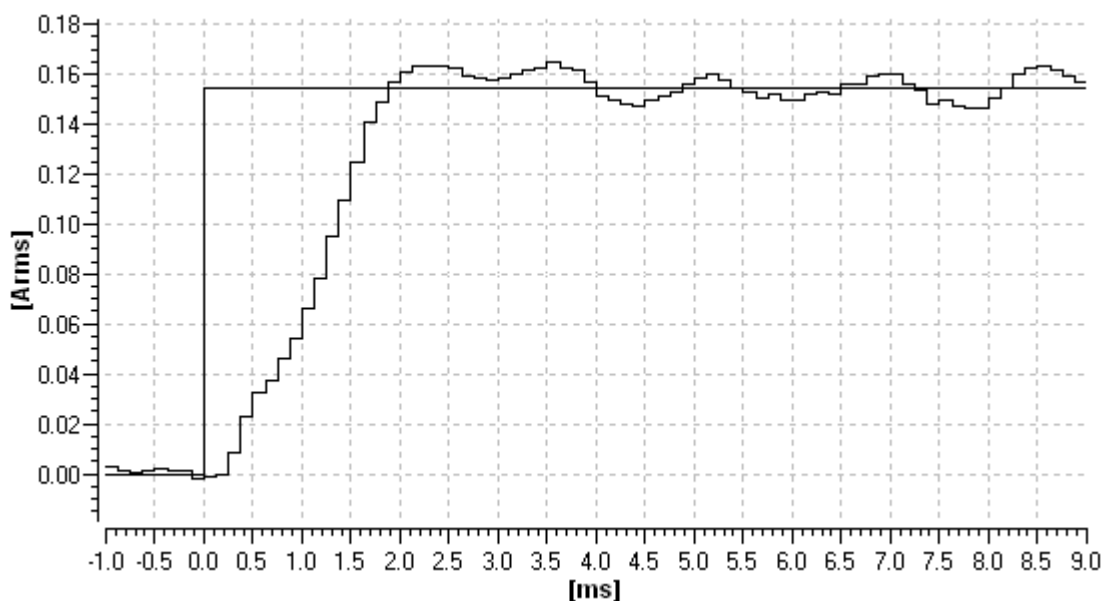
Tab. 1 Parametry regulátorů před optimalizací

číslo osy	regulační smyčka				
	proudová		rychlostní		polohová
	P [V/A]	Ti [ms]	P [Nm.s/rad]	Ti [ms]	P [1/s]
1	9,569	0,55	3,674	13,9	276,8
2	2,3	0,35	7	16,33	97,0
3	8	0,5	1	13,35	208,0
4	10	0,4	0,234	6,85	228,8
5	20,748	1	0,138	4,48	129,1
6	14	0,35	0,2	15,86	106,7

Tab. 2 Parametry regulátorů po optimalizaci



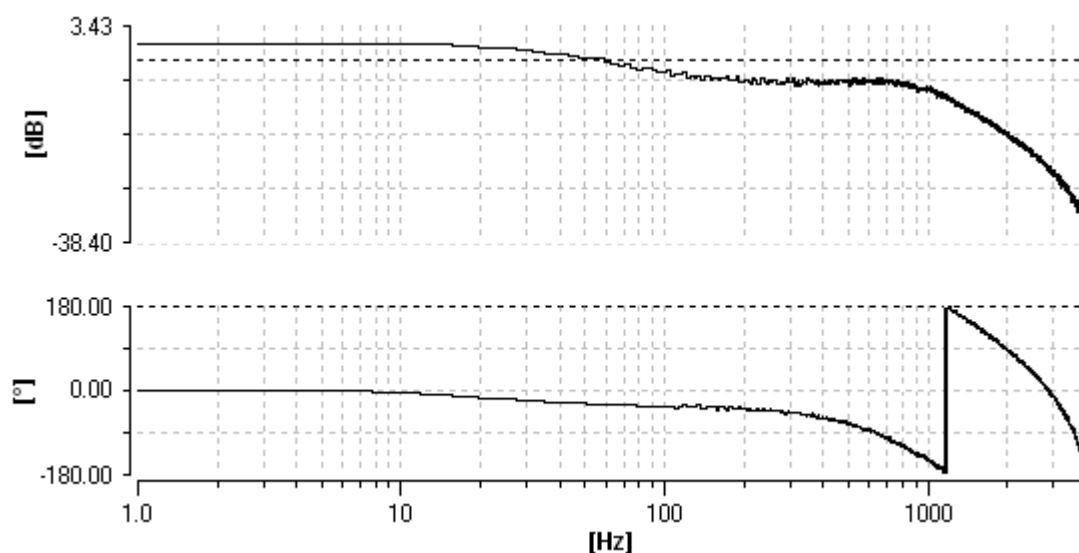
Obr. 6.3 Přejchodová charakteristika proudové smyčky 4 osy před optimalizací



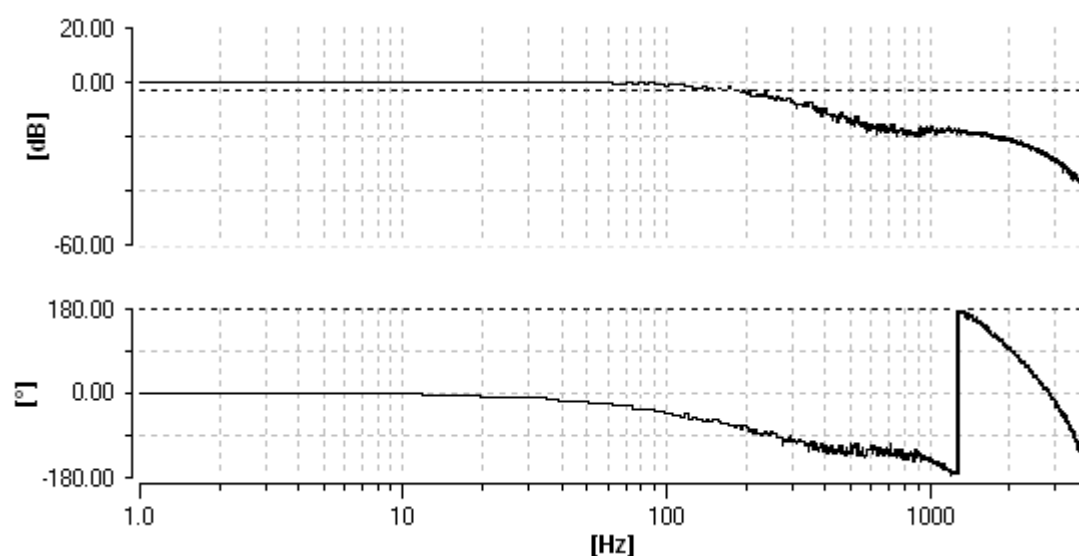
Obr. 6.4 Přechodová charakteristika proudové smyčky 4 osy po optimalizaci

Jednotkový skok byl zvolen jako 2% hodnota z maximálního motorového proudu, který byl u 4 osy nastavena na 7 A. Z porovnání obou přechodových charakteristik (obr. 6.3 a 6.4) je na první pohled zřejmé, že k ustálení regulované veličiny proudu dojde vzájemně za jiný čas. Dobu náběhu lze přibližně považovat v čase 10 ms při nenastavených parametrech proudového regulátoru, kdežto po optimalizaci lze považovat dosažení žádané hodnoty v čase 5 ms. Jak bylo zmíněno proudová zpětná vazba by tedy měla být ze všech smyček v celé kaskádě co nejrychlejší. To dokazuje i průběh přechodové odezvy vůči otáčkové.

Dále byla naměřena frekvenční charakteristika jednak před nastavením, která je znázorněna v obr. 6.5 a po nastavení, jejíž průběh je na obr. 6.6. Ta je složena z amplitudového i z fázového průběhu. Čárkovaná vodorovná čára u amplitudových charakteristik znázorňuje mez -3dB. V průsečíku této čáry s průběhem se nachází mezní frekvence. Měřením proudového regulátoru bylo zjištěno že po optimalizaci je doba ustálení přechodového děje kratší a frekvenční pásmo širší. Před nastavením byla mezní frekvence 50 Hz, po správném nastavení jsme se dostali až na hodnotu převyšující 100Hz.



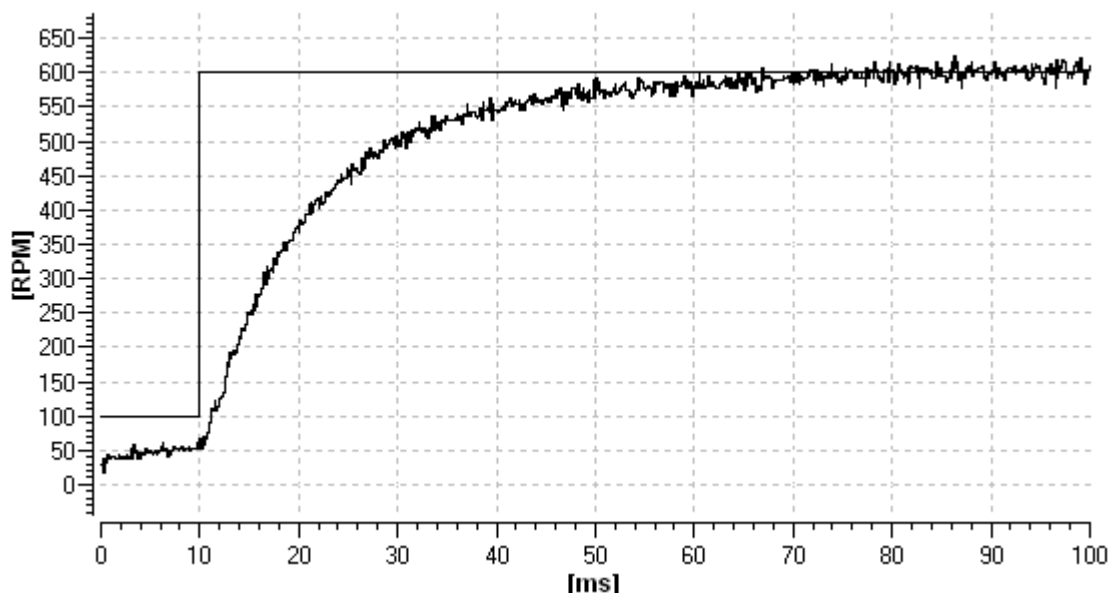
Obr. 6.5 Frekvenční charakteristika proudové smyčky 4 osy před ladící procedurou



Obr. 6.6 Frekvenční charakteristika proudové smyčky 4 osy po ladící proceduře

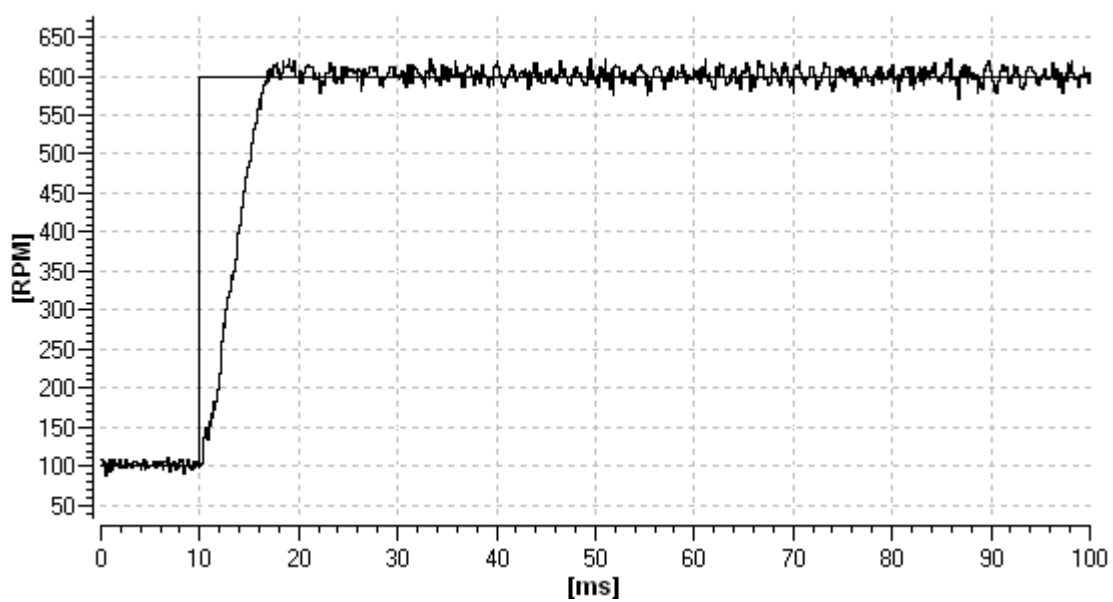
Po optimalizaci proudové regulační smyčky jsme přistoupili na měření otáčkového PI regulátoru. To probíhalo stejným způsobem jako u proudového, tj. přenastavování konstant určených z automatických identifikačních funkcí jsme prováděli až do doby, kdy regulační pochod přestal být stabilní. Na obr. 6.7 je zakreslen průběh přechodové charakteristiky před nastavením, na obr. 6.8 po optimalizaci. Jednotkový skok byl stanoven na 100 ot/min a dále v čase 10 ms na 600 ot/min. I v tomto případě došlo díky ladícím procedurám k lepším přechodovým odezvám za

podstatně nižší čas a k nárůstu šířky frekvenčního pásma. Z obr. 6.8 je patrné, že přechodový děj nedosáhne žádané hodnoty během první části jednotkového skoku. Za čas 10 ms dosáhne rychlost sotva 50% žádané hodnoty, což je způsobeno nevhodným nastavením parametrů P a I.

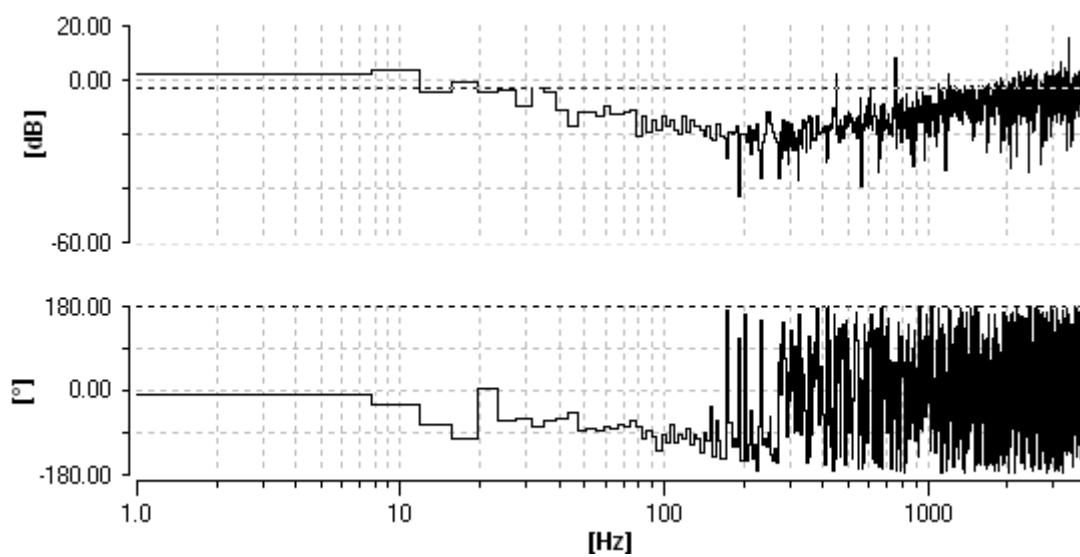


Obr. 6.7 Přechodová charakteristika rychlostní smyčky 4 osy před optimalizací

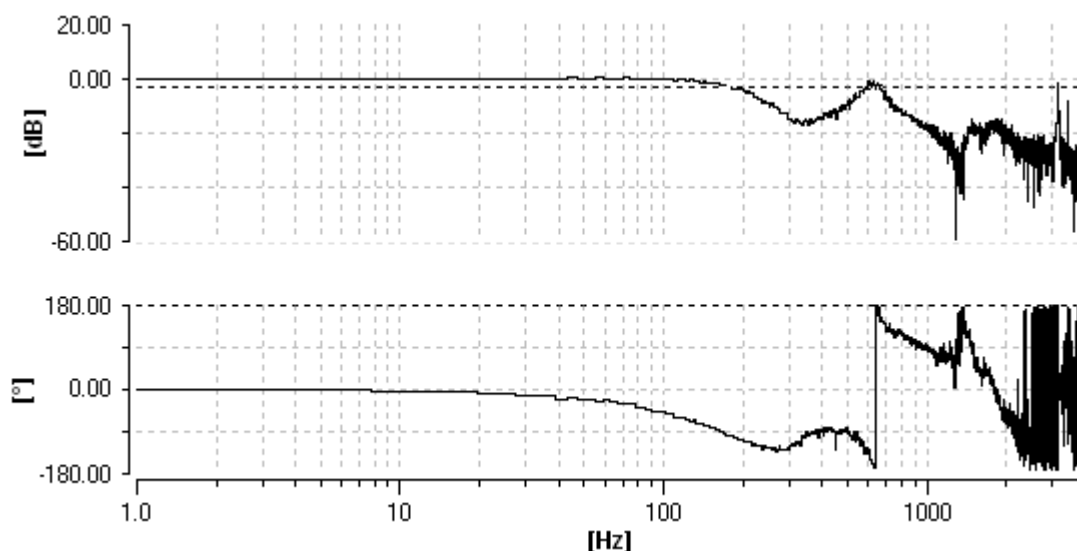
Na obr. 6.9 a 6.10 jsou znázorněny frekvenční charakteristiky rychlostního regulátoru u kterého jsme použili přídatné bloky, které účinně filtrovali vyšší harmonické složky. Tyto bloky představují dané filtry typu DP, HP, aj., které jednotka S120 umožňuje aplikovat a tím podstatně ovlivňovat kvalitu regulace. Tyto funkce umožňuje konfigurovat přímo Simotion Scout. Regulovaná veličina rychlostní smyčky se po nastavení parametrů ustálila na skokovou změnu žádané hodnoty za menší čas a průběh frekvenční charakteristiky se posunul na vyšší mezní frekvenci, jako tomu bylo u proudové smyčky. Fázové charakteristiky porovnávat příliš nelze, nicméně po optimalizaci se mezní fázový kmitočet pohybuje na hodnotě 650 Hz, u kterého se fáze výstupu změní do kladné hodnoty.



Obr. 6.8 Přechodová charakteristika rychlostní smyčky 4 osy po optimalizaci



Obr. 6.9 Frekvenční charakteristika rychlostní smyčky 4 osy před ladící procedurou



Obr. 6.10 Frekvenční charakteristika rychlostní smyčky 4 osy po ladící proceduře

Přechodové a frekvenční charakteristiky zbývajících pěti os byly naměřeny s podobnými průběhy jako tomu bylo u 4 osy. Optimálně nastavené parametry zmenší časové ustálení přechodového děje a zvýší také mezní kmitočet, tudíž se zvětší propustnost frekvenčního pásma. Na tomto místě je třeba připomenout, že měření všech synchronních motorů bylo provedeno přímo na mechanismu, čili se zatíženými hřídeli motorů, jak již bylo výše zmíněno. To jest každá osa musela hýbat s určitou hmotou, která se měnila při různých pracovních polohách. Tyto faktory mohly potom ovlivnit naměřené charakteristiky. U některých rotačních os nastávala situace, kdy navržené parametry proudového a rychlostního regulátoru nevyhovovali, protože konstanty byly optimální jen pro určité polohy kinematických členů jak bylo zmíněno. V těchto pozicích docházelo k rozkmitání regulačního pochodu, což se projevilo jako nepříjemné zvukové vibrace celé mechaniky osy. Tento faktor by se zřejmě podařilo odstranit použitím adaptivních regulátorů pro které by se v řídicím algoritmu počítaly hodnoty parametrů regulátorů. Toto však bylo nad rámec diplomové práce. Proto byly nalezeny takové hodnoty regulátorů, jenž nezpůsobovali jejich nestabilitu v celém rozsahu pohybu robota. Pro představu o vlastnostech pohonu, který byl měřen jako 4 osa je v tab. 3 uvedeno několik konstrukčních elektrických a mechanických parametrů a v tabulce 4 jsou porovnány dosažené kvality regulace před a po nastavení. Všechny naměřené charakteristiky jsou umístěné na CD ve složce *grafy*. Ty lze otevřít pouze v Simotion Scout.

jmenovité napětí [V]	85
jmenovitý proud [A]	2,1
jmenovité otáčky [ot/min]	4000
jmenovitý výkon [kW]	0,42
jmenovitý moment [Nm]	1,2

Tab. 3 Parametry synchronního motoru 4 osy

	proudová smyčka		otáčková smyčka	
	před	po	před	po
doba ustálení [ms]	9	3	70	20
propustné pásmo [Hz]	50	110	20	110

Tab. 4 Porovnání regulačních vlastností před a po ladící proceduře

7. Závěr

Cílem diplomové práce bylo navrhnout otestovat a implementovat algoritmus pro řídicí jednotku průmyslového robota KUKA VK 10/15. Pro tyto účely bylo využito programovacího prostředí MATLAB ve kterém se ověřili výpočty základní teorie přímé a inverzní úlohy robotiky. Dále byla provedena jeho konverze do vývojového prostředí Simotion Scout, který je určený pro tvorbu algoritmů řídicí jednotky D435. D435 spolu s jednotkou S120 komunikuje po sběrnici PROFIBUS a obě hardwarové části tvoří modulární celek, který řídí pohyby robota. Tento systém splňuje víceúrovňový model řízení, který byl popsán v kapitole 3. Ovládání pohybů je možné pomocí PC, tak i pomocí mobilního panelu 177DP. Polohu a orientaci koncového bodu lze určit v kartézském souřadném systému, nebo je možné zadat úhlová natočení jednotlivých rotačních os.

Při vývoji algoritmu jsme vyřešili několik nesrovnalostí, které jsou dostatečně popsány v kapitole 5.6. Jednalo se o matematické výpočty, které oba vývojové nástroje počítali s jinou přesností. Tyto faktory způsobovali nepřesnosti při výpočtech jiných řešení v určitých polohách mechanismu a cílem bylo jejich vzájemné ztotožnění. Jak bylo zmíněno v kapitole 5.6, značné chyby zanesla do výpočtů nedostatečně přesná definice Ludolfova čísla. V dalším kroku bylo potřeba dosáhnout také stejných výsledků při přímkovém řízení polohy. Bylo zjištěno, že na nepřesnosti pohyby robota po přímce má vliv nedostatečné seřízení regulační struktury servopohonů. Proto bylo nutné celou kaskádní strukturu vyladit.

Jednalo se o identifikace a nastavení parametrů proudových, rychlostních a polohových regulátorů. Důvodem bylo zajistit co nejlepší dynamické odezvy jednotlivých os robota zejména při lineárních interpolacích. Tato úloha spočívala v naměření přechodových a frekvenčních charakteristik při výchozích parametrech regulátorů, které software zvolil při konfiguraci hardware. Následovalo použití automatických ladících procedur, jejichž nastavení byla experimentálně doladěna pomocí modifikované Z-N metody pro syntézu regulátorů. V závěru byly porovnány výsledné přechodové a frekvenční charakteristiky s původními, které jsou uvedeny v kapitole 6.2. Byl zjištěn nepříznivý vliv změny momentu setrvačnosti celého mechanismu při polohování. Jeho odstranění však přesahovalo rámec DP.

Celkové úsilí věnované v této práci splnilo však naše očekávání. Tento praktický

příklad demonstruje využití algoritmu, který je aplikován již dnes v drtivé většině řídicích jednotek průmyslových robotů, avšak pro běžné uživatele není veřejný. Kompletní hardware firmy Siemens s průmyslovým robotem KUKA VK 10/15 je po této rutinní úloze připravený pro další implementace jiných důležitých algoritmů, které by plnili konkrétní úlohu. Mohlo by se jednat například o definování kruhové interpolace či funkce zajišťující pohyby kolem referenčního nebo jiného bodu v prostoru.

Literatura

- [1] *D4x5 Commissioning and Hardware Installation Manual*, Siemens AG, 03/2007
- [2] *TO Axis Electric / Hydraulic, External Encoder, Function Manual*, Siemens AG, 03/2007
- [3] *SIMOTION ST Structured Text Programming and Operating Manual*, Siemens AG, 03/2007
- [4] *Motion Control, Technology Object Axis*, Siemens AG, 01/2004
- [5] *Simotion Getting started*, Siemens AG, 01/2004
- [6] <http://www.siemens.cz>
- [7] <http://www.automa.cz>
- [8] elektronická nápověda Simotion Scout
- [9] Novotný Michal: Návrh a realizace prostředků pro kalibraci odměřovacího systému průmyslového robota, Semestrální projekt, 2008
- [10] Noskievič Petr: Modelování a identifikace systémů
- [11] Diblík Martin: Elektické pohony pro dynamicky náročné aplikace, autoreferát disertační práce 2006
- [12] Novák Pavel: Úloha rychlostní smyčky při optimalizaci obráběcího stroje s CNC řízením, diplomová práce, 2008
- [13] Doc. Ing. Šolc František, CSc., Ing. Žalud Luděk, Ph.D.: Robotika, Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně
- [14] Chmel Tomáš: Numerické řešení inverzní úlohy robotiky, Bakalářská práce, 2007
- [15] Lindr David, Diblík Martin: Realizace otevřeného řídicího systému průmyslového robota, EPVE 2008
- [16] Švarc Ivan: Základy automatizace, VUT Brno, Fakulta strojního inženýrství, 2002

Příloha 1. Naměřená data

matice orientace	č.ř.	A1	A2	A3	A4	A5	A6	M	P
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	1	0	0	0	0	-180	0	A	A
	2	0	0	180	0	0	0	A	A
	3	77,69	-152,83	0	75,13	-180	77,69	A	A
	4	77,69	-152,83	180	-75,13	0	77,69	A	A
$\begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	1	0	0	0	0	0	0	A	A
	2	0	0	0	180	0	-180	A	A
	3	0	77,69	-152,83	0	75,13	0	A	A
	4	0	77,69	-152,83	180	-75,13	-180	A	A
$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$	1	0	68,73	-113,82	0	-134,91	-180	A	A
	2	0	68,73	-113,82	180	134,9	0	A	A
$\begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$	1	0	68,73	-113,82	0	-134,91	0	A	A
	2	0	68,73	-113,82	180	134,91	-180	A	A
$\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	1	0	11,83	0,11	0	-101,94	0	A	A
	2	0	11,83	0,11	180	101,94	-180	A	A
	3	0	89,64	-152,94	0	-26,7	0	A	A
	4	0	89,64	-152,94	180	26,7	-180	A	A
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	1	0	11,83	0,11	0	-101,94	-180	A	A
	2	0	11,83	0,11	180	101,94	0	A	A
	3	0	89,64	-152,94	0	-26,7	-180	A	A
	4	0	89,64	-152,94	180	26,7	0	A	A

matice orientace	č.ř.	A1	A2	A3	A4	A5	A6	M	P
$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	1	0	21,01	-39,19	0	108,17	-180	A	A
	2	0	21,01	-39,19	180	-108,7	0	A	A
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	1	0	21,01	-39,19	0	108,17	0	A	A
	2	0	21,01	-39,19	180	-108,7	-180	A	A
$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	1	0	11,83	0,11	0	-101,94	-90	A	A
	2	0	11,83	0,11	180	101,94	90	A	A
	3	0	89,64	-152,94	0	-26,7	-90	A	A
	4	0	89,64	-152,94	180	26,7	90	A	A
$\begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	1	0	11,83	0,11	0	-101,94	90	A	A
	2	0	11,83	0,11	180	101,94	-90	A	A
	3	0	89,64	-152,94	0	-26,7	90	A	A
	4	0	89,64	-152,94	180	26,7	-90	A	A
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	1	0	21,01	-39,19	0	108,17	90	A	A
	2	0	21,01	-39,19	180	-108,7	-90	A	A
$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$	1	0	21,01	-39,19	0	108,17	-90	A	A
	2	0	21,01	-39,19	180	-108,7	90	A	A
$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$	1	7,67	14,94	-16,8	89,74	-97,66	-91,96	A	A
	2	7,67	14,94	-16,8	-90,26	97,66	88,04	A	A
	3	7,67	75,4	-135,95	83,31	-93,76	-150,77	A	A
	4	7,67	75,4	-135,95	-96,69	93,76	29,23	A	A

matice orientace	č.ř.	A1	A2	A3	A4	A5	A6	M	P
$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$	1	7,67	14,94	-16,88	89,74	-97,66	88,04	A	A
	2	7,67	14,94	-16,88	-90,26	97,66	-91,96	A	A
	3	7,67	75,4	-135,9	83,31	-93,76	29,23	A	A
	4	7,67	75,4	-135,9	-96,76	93,76	-150,77	A	A
$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix}$	1	-7,67	14,94	-16,99	-89,74	-97,66	-88,04	A	A
	2	-7,67	14,94	-16,99	90,26	97,66	91,96	A	A
	3	-7,67	75,4	-135,95	-83,31	-93,76	-29,23	A	A
	4	-7,67	75,4	-135,95	96,69	93,76	150,77	A	A
$\begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}$	1	-7,67	14,94	-16,88	-89,74	-97,66	91,96	A	A
	2	-7,67	14,94	-16,88	90,26	97,66	-88,04	A	A
	3	-7,67	75,4	-135,95	-83,31	-93,76	150,77	A	A
	4	-7,67	75,4	-135,95	96,69	93,76	-29,23	A	A
$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	1	7,67	14,94	-16,88	89,74	-97,66	-178,04	A	N
	2	7,67	14,94	-16,88	-90,26	97,66	178,04	A	A
	3	7,67	75,4	-135,95	83,31	-93,76	-119,23	A	N
	4	7,67	75,4	-135,95	-96,69	93,76	119,23	A	A
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}$	0	-	-	-	-	-	-	A	A
$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$	0	-	-	-	-	-	-	A	A

matice orientace	č.ř.	A1	A2	A3	A4	A5	A6	M	P
$\begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$	1	-7,67	14,94	-16,88	-89,74	-97,66	178,04	A	N
	2	-7,67	14,94	-16,88	90,26	97,66	-178,04	A	A
	3	-7,67	75,4	-135,95	-83,31	-93,76	119,23	A	N
	4	-7,67	75,4	-135,95	96,69	93,76	-119,23	A	A
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	1	0	0	0	0	0	90	A	A
	2	0	0	0	180	0	-90	A	A
	3	0	77,69	-152,83	0	75,13	90	A	A
	4	0	77,69	-152,83	180	-75,13	-90	A	A
$\begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$	1	0	68,73	-113,82	0	-134,91	-90	A	A
	2	0	68,73	-113,82	180	134,91	90	A	A
$\begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$	1	0	68,73	-113,82	0	-134,91	90	A	A
	2	0	68,73	-113,82	180	134,91	-90	A	A
$\begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}$	1	0	0	0	0	0	-90	A	A
	2	0	0	0	180	0	90	A	A
	3	0	77,69	-152,83	0	75,13	-90	A	A
	4	0	77,69	-152,83	180	-75,13	90	A	A